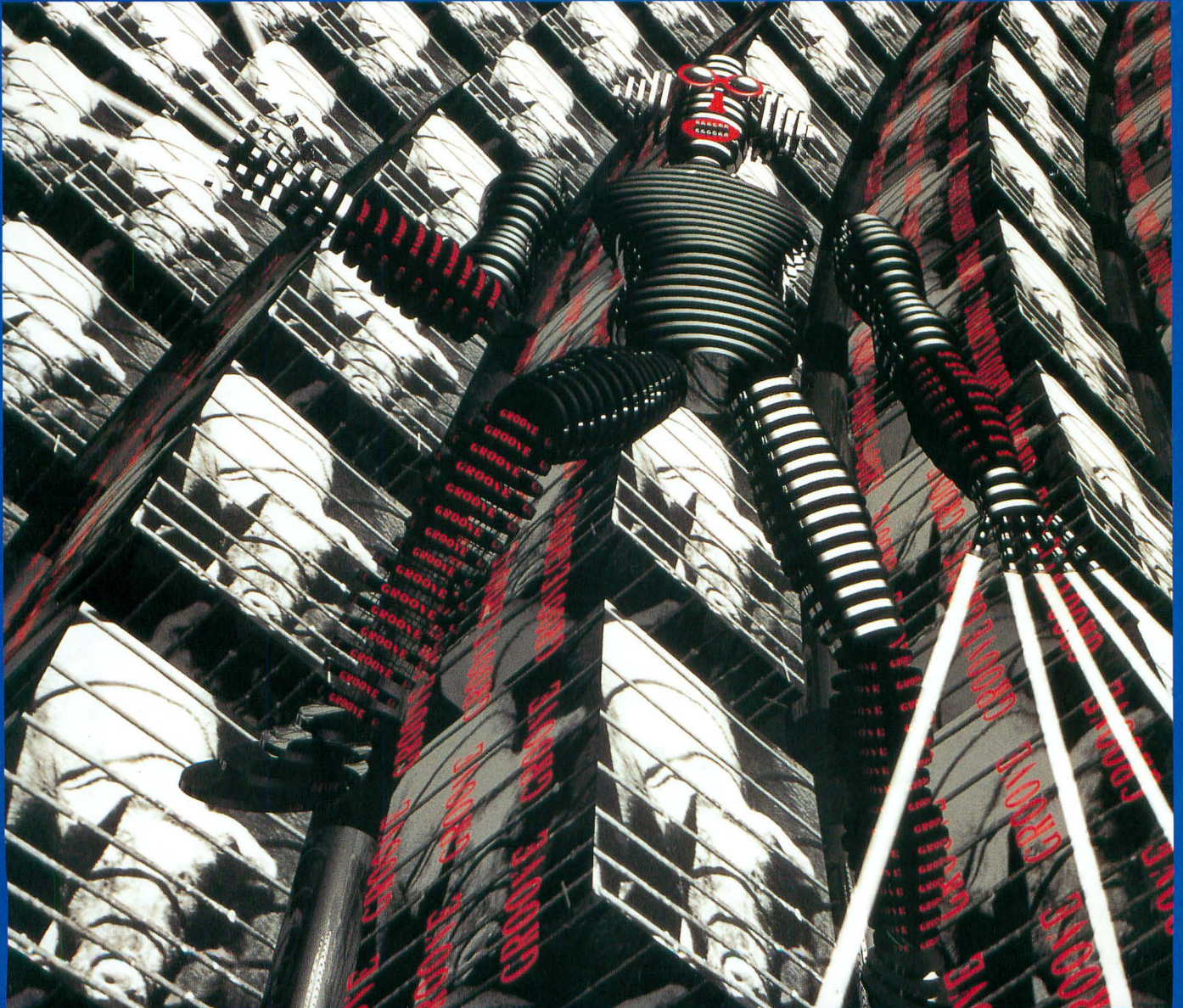
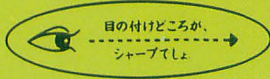
1994年10月1日発行（毎月1回1日発行）第13巻10号通巻150号 昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

特別付録5"2HD もみじ狩りPRO-68K

SX-BASIC/SX-BASIC用ゲーム作成キット/SX-WINDOW用各種ツール
X-BASIC用外部関数/SLASH用モデラMOD.X ver.2.0/タブレットマウスドライバ
TeX入門講座〜てふてふらてふ〜/新製品紹介F-Card V5

10
1994

SOFT BANK オー・エックス
特別定価900円

フィールドが、膨らむ。

先が、ますます面白くなる。

●

未来への確かなビジョンをベースに
発展性のあるプラットホームとしてのウィンドウ環境を提供する
国産オリジナルウィンドウシステムSX-WINDOW。

●

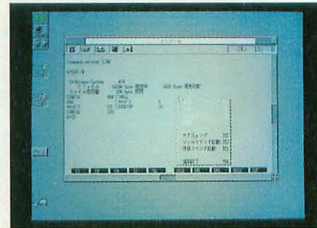
GUI環境や操作環境、高速化へのゆるぎない探求、
マルチメディアの統合的なハンドリング。

●

いま、より多彩なフィールドへ
そのインテリジェンスが展開を始める。

●

次のステージが見えてくる。

●インライン入力のサポート：ASK68K Ver.3.0を利用したインライン入力をSX-WINDOWで実行可能。またシャーペン.Xをワープロとして利用できるよう、さまざまな機能が付加されています。

●コンソールをサポート：Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。
（グラフィックを利用したものなど、SX-WINDOWと処理が重複するものは実行できません。）

●多彩なプリンタに対応：さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ESC/Page、LIPS III、PostScriptに対応したプリンタが利用できます。

SX-WINDOW ver.3.1の
データ利用環境

今も、先も楽しめる。
いつも新展開の予感、SX-WINDOWのニューバージョン。

「SX-WINDOW ver3.1システムキット」CZ-296SS（130mmFD）/CZ-296SSC（90mmFD） 標準価格22,800円（税別）

EXEディスク2プレゼント

シャーペンカスタマイズコンテストの力作や、新作SX-WINDOWソフトウェア情報などを満載のディスク情報誌「EXEディスク2」をプレゼントいたします。

●官製ハガキに住所、氏名、EXE会員番号と90mm（3.5型）/120mm（5.25型）の種別を明記の上、お申込み下さい。また、これからEXEクラブへ入会される方は、商品同梱のEXEクラブ入会申込書に「EXEディスク2希望」と明記の上、ご投函下さい。

応募/問合せ先

〒545 大阪市阿倍野区長池町22-22 シャープ株式会社電子機器事業本部システム機器営業部EXEクラブ事務局EXEディスク2係宛
（TEL 06-621-1221大代表）

申込締切

平成6年11月末日消印有効

発送開始

10月20日より順次発送いたします。

●EXEクラブに入っておられない方は、ソフトベンダー「TAKERU」での購入が可能です。
（平成6年11月1日より2ヶ月間、予価200円）

C O N T

特別企画　もみじ狩りPRO-68K

スーパーストリートファイターII

スターラスター

クイーン・オブ・デュエリスト外伝α+

F-Card V5 for x68k

(で)のショートプロぱーてぃ

●特別企画

●カラー紹介

●シリーズ全機種共通システム

〈スタッフ〉
●編集長／前田 徹 ●副編集長／植木章夫 ●編集／山田純二 豊浦史子 高橋恒行 ●協力／有田隆也 中森 章 林 一樹 吉田幸一 華門真人 朝倉祐二 大和 哲 村田敏幸 丹 明彦 三沢和彦 長沢淳博 司馬 護 清瀬栄介 石上達也 柴田 淳 瀧 康史 横内威至 進藤慶到 ●カメラ／杉山和美 ●イラスト／山田晴久 江口響子 高橋哲史 川原由唯 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 ADGREEN ●校正／グループごじら

表紙絵：塚田 哲也

E N T S

1994 OCT.
10

■広告目次

ビデオグラフィックスの
世界へ。
資料請求券
X68030
Oh!X
10係
■お問い合わせは… シャープ株式会社
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

1,677万色対応、ビデオ映像を高画質・高速取り込み

テレビやビデオ、ビデオディスクなどの映像をX68シリーズやMacシリーズ※1の動画・静止画データとして高速取り込みが可能、いわば“ビデオスキャナ”とでも呼びたいビデオ入力ユニットです。1,677万色対応、最大640×480ドットの高解像度※2。動画・静止画の手軽なハンドリングが、新たなグラフィックシーンを創造します。

※1 MacintoshはIIシリーズ以降の機種に対応、ディスプレイ解像度が640×480ドットの場合、取り込み可能な範囲は、160×120ドット、320×240ドットのサイズになります。
※2 X68030/X68000シリーズでは、1,677万色はデータ作成のみに対応。表示は最大65,536色、解像度は512×512ドット。また、Macintoshは機種により表示色数が異なります。

アプリケーションツール「ライブスキャン」を標準装備

動画や静止画を簡単に保存できるアプリケーションソフト「ライブスキャン」※を標準装備。取り込んでいる映像を表示したり、残したいシーンを簡単に静止画保存したり、手軽な動画・静止画ハンドリングでパソコンの可能性をさらに広げます。X68030/X68000シリーズ用SX-WINDOW対応版とMacintoshシリーズ用QuickTime対応版の2種類を同梱しています。

※SX-WINDOW版はバージョン3.0以降(メモリー4MB以上)、QuickTime版はMacintosh漢字Talk7リリース7.1以上のシステムとQuickTime1.5以上(メモリー8MB以上)が必要です。

1,677万色対応の高速映像取り込み、動画・静止画の手軽なハンドリングが、新たなマルチメディアシーンを創造する。

■SCSIインターフェイス採用：パソコンの専用I/Oスロットを使わずに接続可能になり、汎用化を実現しました。またSCSI-2(FAST)インターフェイスの採用により、データ転送速度の高速化を図っています。X68030/X68000シリーズでは、SCSI-2(FAST)対応のハードディスクを接続することにより、パソコン本体を経由しないで、ハードディスクに直接、動画データをテンポラリデータとして記録することが可能です。パソコン本体のハードディスクへは、記録終了後に、テンポラリデータを変換し動画データとして保存できます。

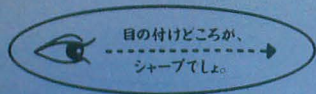
※CZ-600C/601C/611C/602C/612C/652C/662C/603C/613C/653C/663Cに接続する場合は別売のSCSIインターフェイスボードCZ-6BS1ならびにSCSI変換ケーブルCZ-6CS1が必要です。※CZ-604C/623C/634C/644Cに接続する場合は、別売のSCSI変換ケーブルCZ-6CS1が必要です。
※Macintosh Power Bookシリーズに接続する場合は別売のSCSIケーブルなどが必要です。詳しくはMacintosh Power Bookシリーズの取扱説明書をご覧ください。

■高機能MPUを搭載：クロック周波数25MHzの32ビットMPU/MC68EC020を搭載、高速処理やパソコン本体の負担の軽減を実現します。

●MacはMacintoshの略称です。●Macintosh、MacintoshIIは、米国アップルコンピュータ社の登録商標です。●Power Bookは米国アップルコンピュータ社の商標です。●漢字Talk7はアップルコンピュータジャパン社の商標です。●QuickTimeは、米国アップルコンピュータ社の商標です。●価格には、消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は含まれておりません。

ビデオ入力ユニット

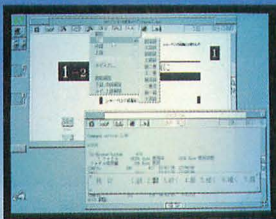
CZ-6VS1

標準価格178,000円(税別)

●さらに実用的なウィンドウシステムへの進化
SX-WINDOW ver3.1 システムキット

CZ-296SS(130㎜FD)/CZ-296SSC(90㎜FD) 標準価格22,800円(税別)

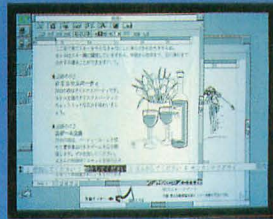
ASK68K Ver3.0を利用したインライン入力のサポート、Human68k/BASICコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できるコンソールのサポートをはじめ、シャーペン.Xをワープロとして利用できるよう機能アップ。また、さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ドローデータ(FSX)/フォントデータ(IFM)処理の高速化も実現しています。

※コンソールでは、SX-WINDOWと処理が重複するものは実行できません。

4MB

●定評のGUI対応ウィンドウワープロ
EGWord SX-68K

CZ-271BWD 標準価格59,800円(税別)

ウィンドウワープロとして評価の高いEGWordのSX-WINDOW対応版。キャラクタベースのワープロを超えたグラフィカルユーザーインターフェイス(GUI)による手軽なDTPソフトとしても優れた表現力を発揮します。定評ある日本語入力方式(EGConvert)によるインライン入力、さまざまなグラフィックデータ(GScript)やテキストデータの貼り込み、また文書互換を実現するEDF(Extended Document Format)形式をサポートしています。

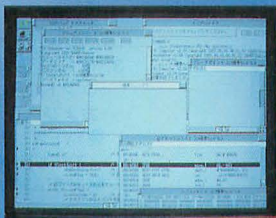
4MB、ver.2.0

※5MB以上の空きのあるハードディスクが必要です。

●SX-WINDOW開発支援ツール
SX-WINDOW 開発キット Workroom SX-68K

CZ-288LWD 標準価格39,800円(税別)

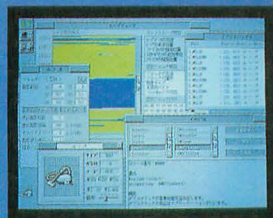
SX-WINDOW用のソフト開発に必要なツールやサンプルプログラムを装備。プログラムの編集、リソースの作成、コンパイル、デバッグといった一連の作業をSX-WINDOW上で効率よく実行できます。初めてSX-WINDOW用のプログラムに挑戦する人にも、簡単に基本機能の理解が深まる33種(基礎編23種、応用編4種、実用編6種)のサンプルプログラム付き。

※ご使用に当ってはC compiler PRO-68K ver.2.1が必要です。

4MB、ver.2.0

●SX-WINDOW開発キットのサポートツール
開発キット用ツール集

CZ-289TWD 標準価格12,800円(税別)

SX-WINDOW開発キットをさらに使いやすくするためのツールです。SXコールの簡易リファレンスを簡単に検索するインサイドSX、イベントの発生を常時監視・確認するイベントハンドラ、リアルタイムにメモリブロックの利用状況を表示するヒープビューアなど11種のツールが用意されています。

4MB、ver.2.0

●SX-WINDOW対応ドローイングツール
Easydraw SX-68K

CZ-264GWD 標準価格19,800円(税別)

イラスト、フローチャート、地図、見取り図など各種グラフィックが製図感覚で作成できます。作成したデータは他のSX-WINDOW対応アプリケーションでも利用でき、企画書などの作成をサポート。ページプリンタドライバも標準装備。

4MB、ver.3.0

●ウィンドウ対応グラフィックツール
Easypaint SX-68K

CZ-263GWD 標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応ペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。

2MB、ver.1.1

●SX-WINDOWを楽しく使うためのアクセサリ集
SX-WINDOW デスクアクセサリ集

CZ-290TWD 標準価格14,800円(税別)

SX-WINDOWをさらに便利に楽しく使うためのデスクアクセサリ集です。スクリーンセーバ、スクラップブック、スケジューラ、アドレス帳、電子手帳通信ツール、パズルなど、12種の豊富なアクセサリが収められています。

4MB、ver.3.0

●マルチタスク機能をはじめ通信環境がさらに充実
Communication SX-68K

CZ-272CWD 標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションを実行中でも簡単に通信が可能。自動ログイン機能やプログラム機能、など豊富な機能をサポートしています。

2MB、ver.1.1

●FM音源サウンドエディタ
SOUND SX-68K

CZ-275MWD 標準価格15,800円(税別)

他のミュージックソフトで演奏中の音色を、簡単に作成、変更できるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。

2MB、ver.1.1

●SX-WINDOW対応になってさらにパワーアップ
倉庫番リベンジ SX-68K ユーザー逆襲編

CZ-293AW(130mmFD)/CZ-293AWC(90mmFD) 各標準価格6,800円(税別)

倉庫番10年にわたるユーザーの投稿など、新作306面が目白押し。まさに倉庫番の最強版がSX-WINDOW上で楽しめます。AI機能やエディット機能、キャラクタ変更機能も装備。半年で解けたらあなたは天才?です。

2MB、ver.1.1

PRO-68K シリーズ

●X68030/X68000対応

CZ-295LSD 標準価格44,800円(税別)

※メインメモリ2MB以上が必要です。

C compiler PRO-68KのX68030/X68000対応版。MPU68030、MC68882の命令セットに対応したアセンブラ、デバッガ、ソースコードデバッガを付属。またHuman68k ver.3.0、ASK68K ver.3.0にも対応。新たにGPIBライブラリ、MC68882対応フロートライブラリを付属しています。

※ 2MB,ver.1.1 の表示は、メインメモリ2MB以上、SX-WINDOW ver.1.1以上が必要であることを示します。

※発売予定のソフトの画面は実物とは異なる場合があります。

●EGWord、EGConvertは株式会社エルゴソフトの登録商標です。

●お問い合わせは…シャープ㈱電子機器事業本部(液映)システム機器推進プロジェクトチーム 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ シャープ株式会社

資料請求券

SOUND Canvas GS音源対応
MIDIマルチレコーダー

Mu-1 GS
Musicstudio
[ミューワンジーエス]

Mu－1GSはローランド社SC－55mkⅡなどGS音源をフルに活用できるコントロール群と高度な音楽表現を可能にする新感覚エディットウインドウ搭載のMusicstudioプロフェッショナルバージョンです。

スタンダードMIDIファイル オリジナルアーティストシリーズ　各¥3,500

SCB-1001

SCB-1002

PIECES OF WORK
SCB-1003

HOPE
SCB-1004

SCB-1001 duplicity／佐久間正英
SCB-1002 ブレインボックス美術館／国本佳宏
SCB-1003 PICES OF WORKⅡ／本多俊之
SCB-1004 HOPE／松居慶子

■推奨音源：Roland SC-55, SC-55mkⅡ, SC-88
SC-33, CM-500, CM-300

■「GS対応エクスクルーシブデータ」を使用しています。GM音源など推奨音源以外の機器を使用する場合、音量等のバランスが異なりますのでエディットしてお聴きください。

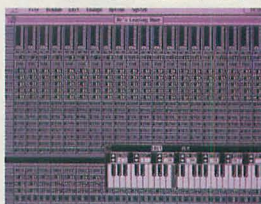
■SC-88対応「スタンダードMIDIファイル クラシックシリーズ」6タイトル発売予定

〈星智輝 with T. T. CDシングル〉
「No Frame No Fame/Christmasday Ring Our Hearts」
定価 ¥1,000（税込）
☆九十九電機パソコン本店Ⅱ4Fで販売しております。どうぞご利用下さい。

特長

1. 新感覚エディットウインドウ
   ●エディタ感覚のプロフェッショナル仕様

2. 簡単エクスクルーシブ入力
   ●チェックサム自動計算入力
   ●曲中でも使用可能！

3. GS音色コントロール機能
   コントロールコードのリアルタイムコントロールおよびステップ入力が可能
   ●TVFカットオフフリケンシー、TVFレゾナンス、TVF&TVA・アタック、ディケイ、リリース・タイム
   ●ドラムインストゥルメント・ピッチ、リバーブセンドコーラスセンド、パンポット、ボリューム

4. RCPコンバート機能追加
   ●カモンミュージックRCM、STED2
5. 24トラック／リアルタイム録音／ステップ入力機能
6. X68030(25MHz)／Human68K Ver.3.01対応
7. RS－232C／MIDI出力対応
   （注意：出力のみ対応、単独使用不可／要MIDIボード）
8. 内蔵FM／ADPCM音源対応
9. 国本佳宏／GS対応デモ曲収録

データコンバート一覧表

読み込み（Load）		ファイル
Mu－1GS←	ミュージ郎／ミュージ郎Ⅱ MUSIC PRO-68K FM&MIDI MML MML スタンダードMIDIファイル フォーマット0/1 RCM／STED2	SNG MUS OPM MID RCP
書き込み（Save）		**ファイル**
Mu－1GS→	スタンダードMIDIファイル フォーマット1 MUSIC PRO-68K FM&MIDI	MID SCO

ハード構成 シャープ68000／030本体
MIDIボード（シャープ社製CZ-6BM1またはシステムサコム社製SX-68M／SX68MⅡ）
ローランド社製GS対応音源SC-55、SC-55mkⅡ SC-300、SC-500など

Mu-1 GS 標準価格 ¥28,000（税別）

■本ソフト動作には、メインメモリ2MBが必要です。

〒213 神奈川県川崎市高津区下作延1043
TEL 044－855－4335

パーソナル・コンピュータゲームマガジン
ゲームブラスト
GAMEBLAST

GAMEBLASTは
ここがうれしい!
1毎号たっぷり
60ページのレビュー!
2パソコン初心者も
すぐ楽しめる親切編集
3これは注目!
海外ソフト最新情報
4要チェック!
充実のランキング
©1993 Embodiment Films,AS Entertainment Planning &PIA
お近くの書店でお買い求め・ご予約ください。
SOFT BANK
ソフトバンク株式会社／出版事業部

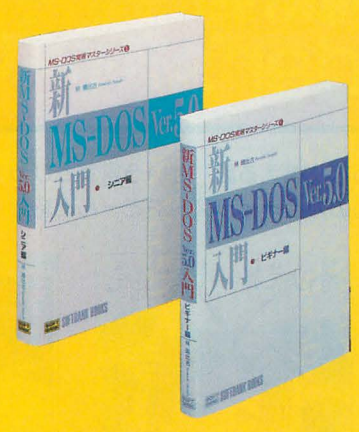
Hello!PC GAMEBLAST

創刊を記念してブックフェアを開催中です。
より深くパソコンを知りたいあなたを
お待ちしています。

ハロー！パソコンフェア開催書店

地区	書店名	期間	電話番号
北海道			
札幌市	東京旭屋書店札幌店	9/2〜	011-241-3007
〃	YES札幌電脳館	9/8〜10/8	011-214-2849
旭川市	YES旭川	9/1〜9/30	0166-26-2251
函館市	YESさいか	9/8〜	0138-27-2551
東北			
仙台市	金港堂ブックセンター	9/1〜9/30	022-223-0979
〃	YES仙台	9/8〜	022-266-6212
〃	パソコンDac仙台東口店	9/1〜12/31	022-291-4744
いわき市	ヤマニ書房本店	9/8〜	0246-23-3481
関東			
宇都宮市	宇都宮ブックセンター	9/1〜10/31	0286-27-7700
千葉市	ラオックスTHECOMPTER館千葉店	9/8〜	043-245-9371
柏市	新星堂柏店	9/8〜	0471-64-8551
東京			
千代田区	書泉グランデ	9/8〜9/30	03-3295-0011
〃	書泉ブックタワー	9/7〜9/30	03-5296-0051
〃	ラオックスTHE COMPUTER館	9/8〜10/7	03-5256-3111
〃	BitINN 東京	9/3〜9/30	03-3255-4575
〃	T-ZONEミナミ店	9/1〜9/25	03-3257-2776
中央区	八重洲ブックセンター本店	9/8〜	03-3281-1811
渋谷区	大盛堂書店	9/5〜9/30	03-3463-0511
〃	東京旭屋書店渋谷店	9/1〜9/30	03-3476-3971
渋谷区	三省堂書店渋谷店	9/1〜9/30	03-3407-4545
新宿区	紀伊國屋書店本店	9/2〜9/30	03-3354-0131
〃	博文堂新宿店	9/1〜	03-5330-3680
〃	ゼロワンショップ新宿店	9/1〜9/30	03-3345-9401
〃	ゼロワンショップ高田馬場店	9/1〜9/14	03-3208-6201
豊島区	新栄堂本店	9/1〜	03-3984-2345
〃	東京旭屋書店池袋店	9/1〜	03-3986-0311
大田区	アクトブックスサンカマタ店	9/22〜	03-3735-1551
国分寺市	三成堂書店国分寺店	9/8〜	0423-25-3211
八王子市	くまざわ書店八王子店	9/1〜9/30	0426-25-1201
〃	ムラウチブック	9/1〜9/30	0426-44-0505
神奈川			
横浜市	有隣堂横浜駅東口ルミネ店	9/8〜9/30	045-453-0811
〃	有隣堂伊勢佐木店	9/1〜9/30	045-261-1231
〃	そごうブックセンター	9/8〜	045-465-2111
〃	丸善ブックメイツポルタ店	9/1〜9/30	045-453-6811
〃	T-ZONE横浜店	9/1〜	045-641-7741
川崎市	文学堂本店	9/1〜9/30	044-244-1251
横須賀市	平坂書房WALK横須賀店	9/8〜10/7	0468-25-5537
厚木市	有隣堂厚木店	9/8〜	0462-23-4111
甲信越・北陸			
新潟市	紀伊國屋書店新潟店	9/10〜9/24	025-241-5281
長野市	平安堂長野店	9/上旬〜	0262-26-4545

ソフトバンク株式会社／出版事業部

〒103 東京都中央区日本橋浜町3-42-3 販売局：TEL.03-5642-8101

新MS-DOS Ver.5.0入門 ビギナー編
定価1,900円

新MS-DOS Ver.5.0入門 シニア編
定価2,400円

新C言語入門
ビギナー編 定価1,900円

新C言語入門
シニア編 定価2,400円

Windows入門 ビギナー編
定価1,900円

一太郎Ver.5 for WINDOWS SUPER BOOK
エントリーユーザ編 定価2,000円

はじめの一歩 Lotus1-2-3 R2.4J
定価1,300円

Lotus1-2-3 R4J SUPER BOOK
定価2,000円

まるごと卓駆☆ブック
定価2,980円

まるごとメモリブック
定価2,980円

EXCEL 5.0 SUPER BOOK
エントリーユーザ編 定価2,400円

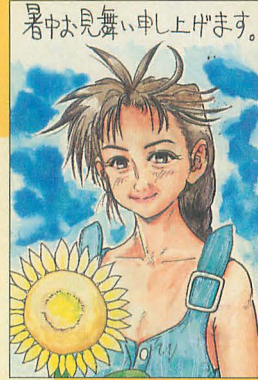
まるごとFDブック
定価2,980円

地区	書店名	期間	電話番号
甲府市	貢川朗月堂	9/1～9/30	0552-28-7356
富山市	清明堂書店	9/1～9/30	0764-24-4166
金沢市	書林香林坊本店	10/10～	0762-20-5011
福井市	勝木書店	9/上旬～	0776-24-0428
東海			
浜松市	浜松谷島屋連尺店	9/8～	053-453-9121
名古屋市	三省堂書店名古屋店	9/8～9/30	052-562-0077
〃	丸善名古屋支店	9/1～9/30	052-261-2251
〃	三洋堂書店上前津店Σ	9/8～	052-251-8334
〃	三洋堂書店いりなか本店	9/8～9/30	052-832-8202
〃	コンプマート名古屋	9/1～12/31	052-581-1241
〃	池下三洋堂	9/8～9/30	052-762-2345
豊橋市	精文館書店本店	9/8～	0532-54-2345
関西			
京都市	アバンティブックセンター	10/1～10/31	075-682-5031
〃	J&P京都寺町店	9/1～9/30	075-341-4411
〃	J&P京都１番館	9/1～9/15	075-323-1511
和歌山市	宮井平安堂	9/1～9/30	0734-31-1331
大阪市	紀伊國屋書店梅田店	9/1～	06-372-5821
〃	旭屋書店本店	9/8～	06-313-1191
〃	駸々堂書店京橋店VERSION99	9/1～9/30	06-353-4011
〃	J&Pテクノランド	10/1～10/15	06-634-1211
〃	J&Pコスモランド	9/1～9/14	06-634-3111
大阪市	ニノックス エレホビー店	9/1～9/30	06-632-2038
〃	ニノックス コア日本橋	9/1～9/30	06-647-2038
神戸市	ジュンク堂三宮店	9/8～9/30	078-392-1001
〃	星電社PC-PORT	9/1～9/30	078-331-1156
〃	星電社ハーバーランド店	9/23～	078-360-8821
〃	PC X TOWN	9/1～9/30	078-391-6356
姫路市	ニノミヤ姫路店	9/1～11/30	0792-88-2363
中国·四国			
広島市	COMCITY	9/1～9/15	082-240-3381
福山市	ダイイチCOMCITY福山	9/1～12/31	0849-21-1200
宇部市	京屋書店	9/1～10/31	0836-31-2323
松江市	ベストコンピュータウンまつえ	9/20～10/10	0852-31-6752
高松市	宮脇書店本店	9/1～	0878-51-3733
九州·沖縄			
福岡市	紀伊國屋書店福岡店	9/8～9/30	092-721-7755
〃	ベストコンピュータウンふくおか	9/8～9/30	092-741-7677
北九州市	ナガリ書店	9/8～9/30	093-521-1044
熊本市	ベストコンピュータウンくまもと	9/8～9/30	096-322-4180
大分市	ベストコンピュータウンおおいた	9/8～9/30	0975-32-9396
宮崎市	ベストコンピュータウンみやざき	9/8～9/30	0985-22-8325
鹿児島市	Booksみすみ南港1号店	9/8～9/30	0992-57-3611
那覇市	球陽堂書房本店	9/8～9/30	098-863-3752

暑中見舞いだ!
Oh!X reader'sぎゃらりぃ

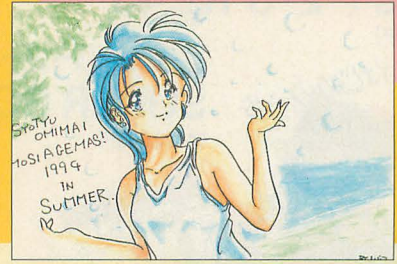
今年の夏は去年の分までやってきたような“燃える夏”という言葉がぴったりでした。そんななか，一服の清涼剤となった皆さんのカラーイラストをどーんと紹介しましょう。

▲板橋　芳則（福島県）

▲大高　孝平（宮城県）

▲占部　哲彦（広島県）

▲青木　一師（奈良県）

▶平　智征（神奈川県）

▲加藤　隆（佐賀県）

▲横井　賢一（富山県）

▲神山　一郎（神奈川県）

▲徳物　信生（広島県）

▲武田　正道（兵庫県）

▲林　寛（埼玉県）

◀前田　基行（兵庫県）

▲玉野　健一（奈良県）

▲日高　光代（宮崎県）

▶鈴木　貴久（神奈川県）

新製品紹介

プロフェッショナル3Dレンダラ

L/Image

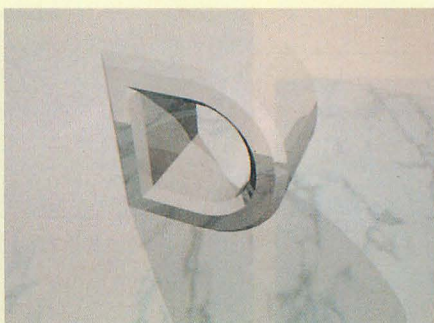
L/Imageとはイマジカテクノシステムが X68000用に発売する3Dレンダリングソフトウェアだ。

聞き慣れない社名かもしれないが，PersonalLinksといえば思い当たる人もいるのではないだろうか。イマジカテクノシステムは，日本のお家芸メタボールの総本山ともいえるソフトハウスである。

ワークステーションでのPersonalLinksはモデラからペイントソフト，アニメーションツールまでを包括した大きなシステムなのだが，L/Imageとはその中枢にあたるレンダリングエンジンのことである。

今回X68000用に発売されるのはこのシステムのレンダラのみ。モデラやアニメーションツールといったシステムは付属しない。ワークステーション版では標準ポリゴンエディタのL/Shapeやメタコーポレーションジャパン製のメタエディタを使う，というのが標準的だったのだが，X68000ではユーザーが直接，イメージスコアという形式のテキストファイルを記述してレンダラに送らなければならない(まあ，C-TRACEなどを使っていた人にはなんでもないことだろうが)。

レンダラのみといっても，このレンダラが半端ではない。グラフィックワークステーション用のレンダラはほかにも多数存在するが，とりわけ出力画像の美しさでは定評がある。CGプロダクションでも，モデリングはほかのツールでやっても最終的なレンダリングはL/Imageを使うことも多いという。

基本的には分散レイトレーシングを行うレイトレーサだが，1次レイをスキャンライン法で探索するのでリーズナブルな速度でレンダリングが可能。基本形状として，平面や各種3次曲面，ポリゴンパッチ，メタボールなどあらゆるものに対応している。見当たらないのはNURBSとパーティクル

くらいか(代わりに(?)ファイアタイプ指定がある)。マッピングはカラーやバンプは当然として，環境マッピングなどにも対応，そのほか，半影，モーションブラー，デフォーカスなど特殊なレンダリングも可能。

●どんなものが作れるのか？

半年くらい前のOh!Xを引っ張り出してみてもらいたい。この頃まで隔月で本誌の表紙を担当していただいた須藤氏が使用していたのがPC-9801版のPersonalLinksシステムである(レンダラはL/ImageではなくTRACY。PC-9801版はポリゴンレンダラとレイトレーサが分離されている)。

そのほか，有名なところではフジF1 Grand Prixのオープニングタイトルあたりが PersonalLinksで作成された作品だ。それらを見ればモデリングの自由度や質感などの表現力がだいたいわかるのではないだろうか。

●DōGA形状データのレンダリング

なお，モデラがない代わりに，Z'sTRIPHONY，DōGA CGAシステムなどからのデータコンバータが付属している。当面はDōGAのRENDに代わって高画質レンダリングを行うという用途が主になるだろうか。

反射，屈折といったアトリビュートを使用していない部分はスキャンラインアルゴリズム(DōGAのRENDと同様のもの)でレンダリングされるので十分な速度が期待される。単に「RENDにバンプマッピングやフォンシェーディングがついたもの」というだけで大歓迎する向きもあるだろう。

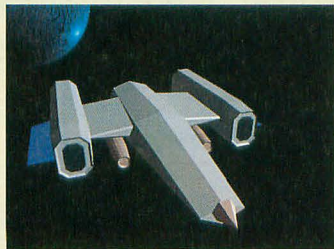
●要求されるシステム

製品にはFLOATドライバを使用したものとX68030+68882に対応した2バージョンの実行ファイルが収録される模様(PC-9801版はコプロ専用しかない)

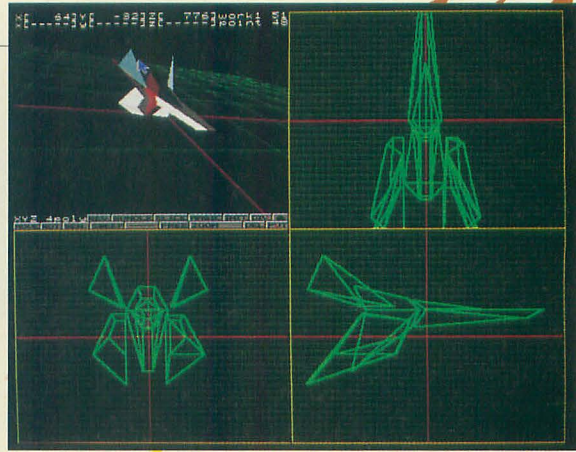
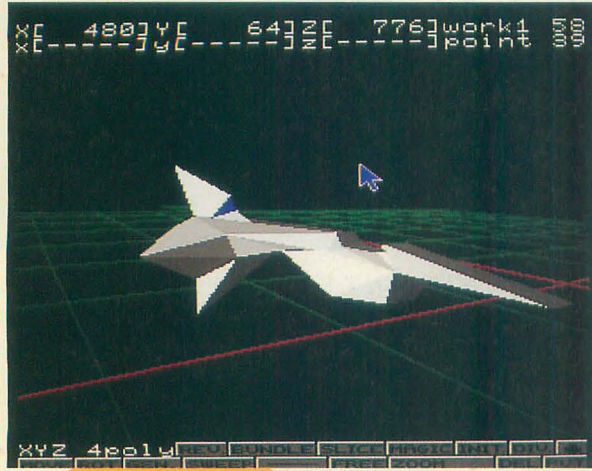
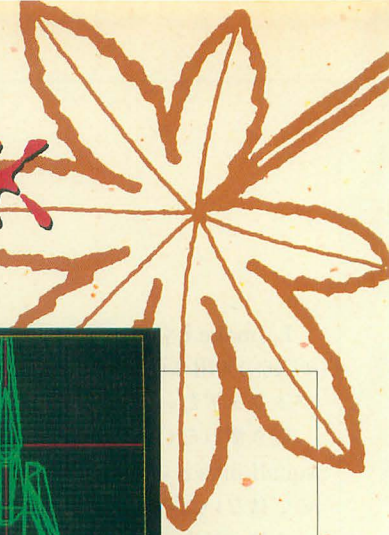
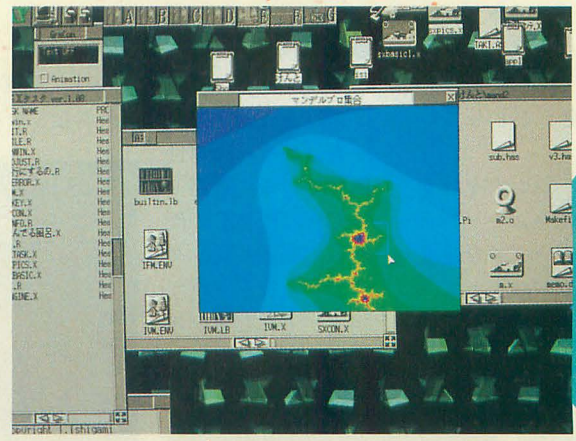
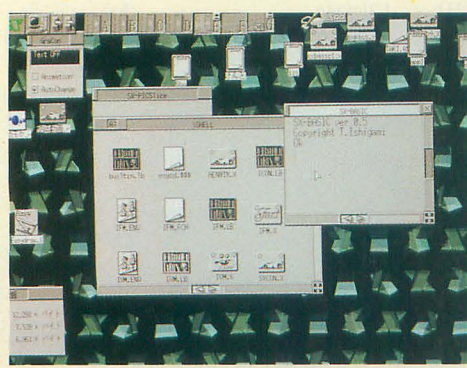
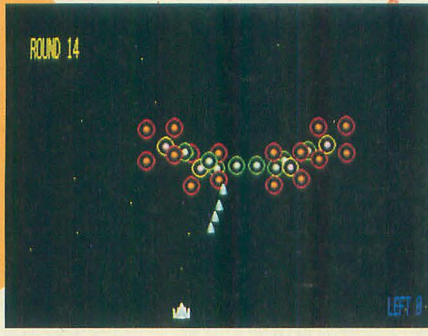
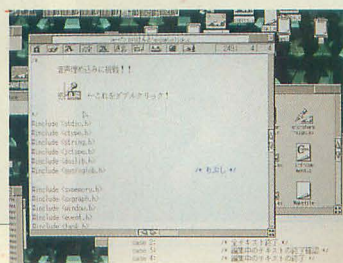
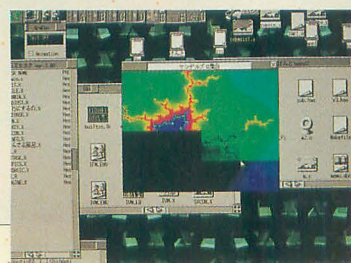
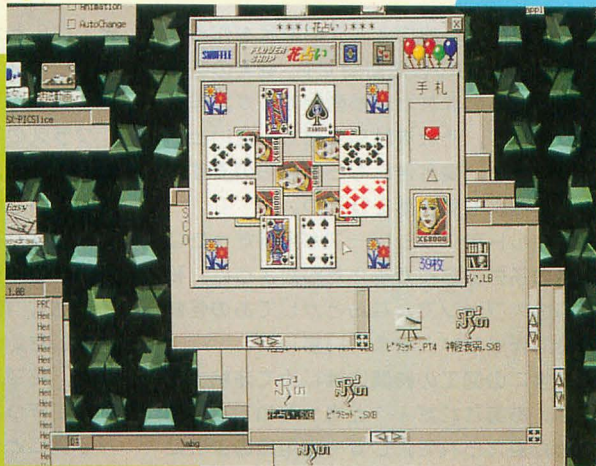
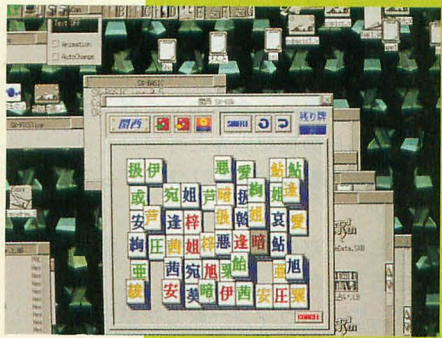
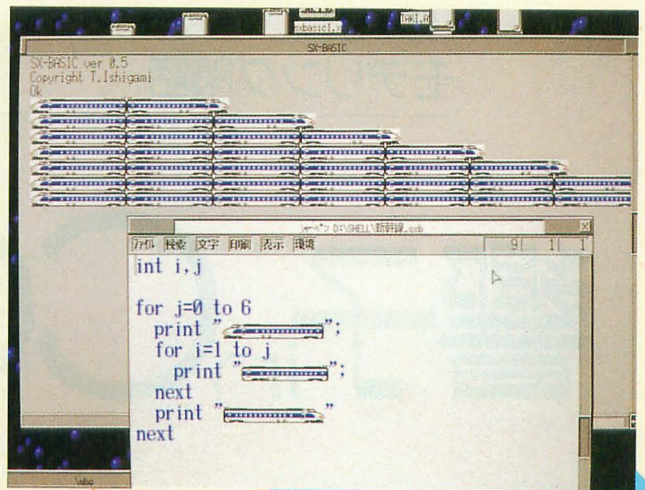
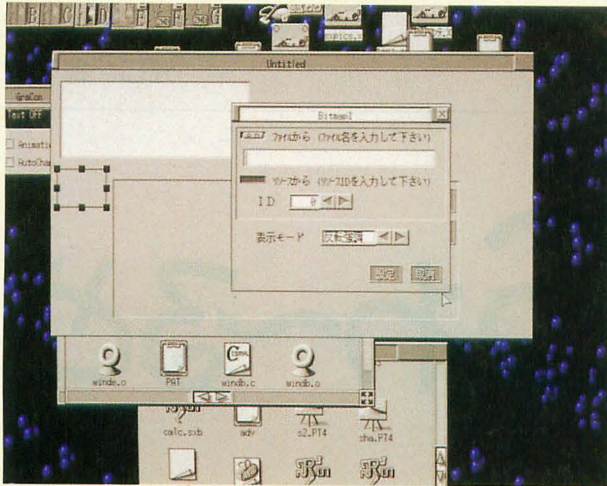
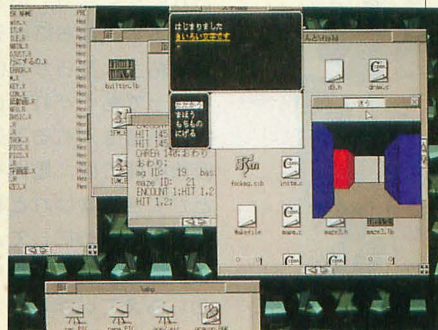
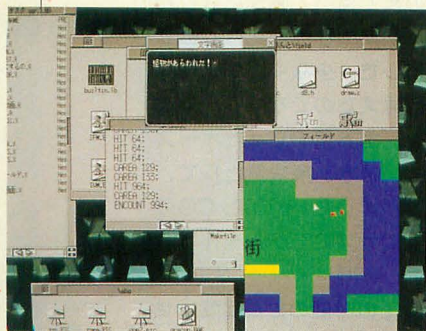
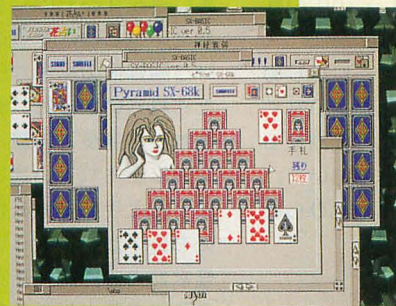
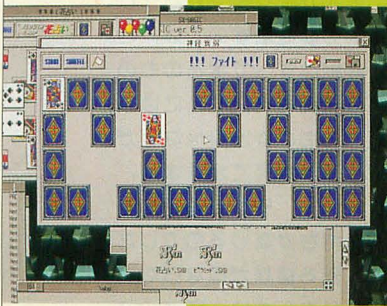
実行ファイル900Kバイトという巨大なシステムであるためメモリは要4Mバイトとなっている。

注目は「DōGAのREND程度を目指す」という実行速度だが，X68030や040turbo使用時のレンダリング速度など興味は尽きない。

X68000版は11月発売予定だ。詳細な情報が入り次第，追って紹介することにしたい。

L/Image　予価58,000円(税別)
イマジカテクノシステム　☎03(5449)3451

［特別企画］

もみじ狩りPRO-68K

秋の夜長に贈る今年4回めの付録ディスクです
ゲームから各種ツールまで，まずは画面をお楽しみください

上は全面的にバージョンアップされたSLASH用モデラMOD.X ver.2.0だ。多少慣れは必要だが，多機能で，かなり操作しやすくなった。右はX-BASICでマシン語を扱うEXEC.FNCのサンプルゲームの画面。タイムアタックが熱い。さらに右はPUSH BON!の追加面データだ。全210面。さらなる難問が待ちかまえているぞ。

SX-WINDOW用の各種ツール。左はXロゴがくるくる回る背景動画のサンプル。マンデルブロ集合表示ツール(テスト版)は任意の範囲を拡大できる。そのほか，下は大幅にバージョンアップされたベル.Xとシャーペンの外部コマンドの一例だ。XCLICKとXPASTEはシャーペン上で「アイコンをクリックすると……」という動作を実現する。

SX-BASIC暫定版その3と仲間たち。本格的にリソースに対応し，マルチフォントテキストにも対応しているので上のようなプログラムも可能。左はSX-WINDOW上で2D，3DタイプのRPGを実現するためのサポートツールだ。表示関係はすべてゲームエンジンに任せ，プログラム本体はSX-BASICで記述できる。下は投稿によるSX-BASIC用カードゲームいろいろ。カードデータは共通リソースで扱っている。

MJ700V2Cでのカラーハードコピー。とりあえず先月のものと見比べてみてほしい。

［第41回］
モデリング閑話

江口　響子

響子inCGわ～るど

モデリングとは，一般に３次元の形状を作ることを意味します。CGの仮想空間でも，現実の空間でも，立体を作るのはなかなか面倒ですが，作りたいものがあって，それをはやく見てみたいという気持ちになると，案外苦にならないものです。

今回はそんなよもやま話をひとつ。

テレビでウルトラマンシリーズが全盛だったころ……当時の子供の多くがそうだったように，私も円谷プロの大ファンで，なんとかしてあの怪獣たちを自分で作りたいと考えていました。

小学校低学年の図工の時間にはじめて油粘土を触ったとき，ああこれだ！　と思ったのです。自由自在に形が変えられて，しかも画用紙のようにぺちゃんこではない，これなら，テレビの中の怪獣そっくりにできると。

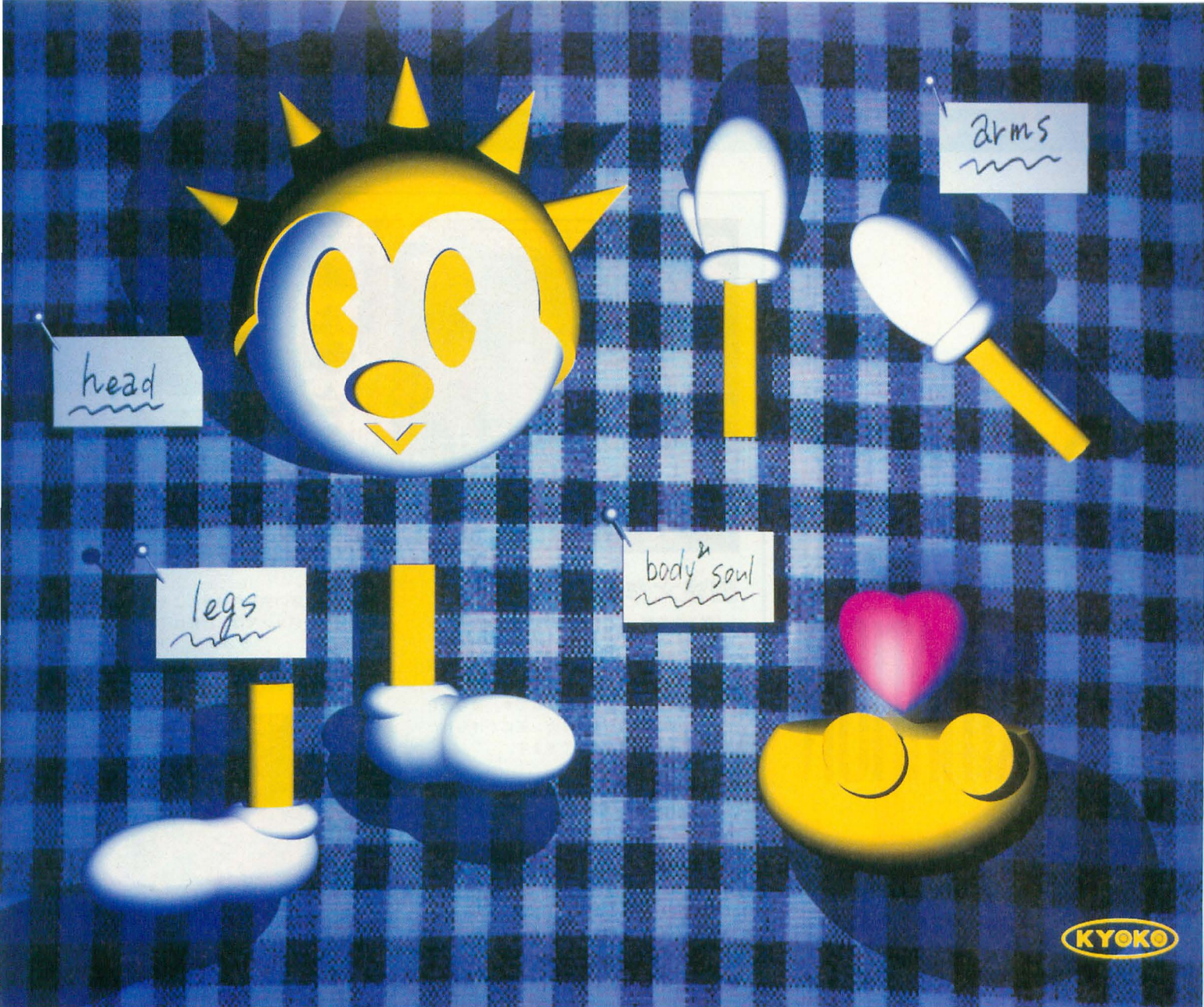
最初のころは，いきなり全身から形作りはじめていましたが，しっぽのところで粘土が足りなくなったりして，うまくいきません。試行錯誤するうちに，粘土を，頭や体，手足と分けておいて，最後に組み合わせればいいんじゃないかとひらめき，なんとか作り上げました。怪獣の種類ははっきりと覚えていませんが，エレキングかピグモンだったと思います。

クラスの同級生の何人かは，かっこいい～とほめてくれて，自分でもなかなかよくできたと悦に入っていたのです。「たいへんよくできました」のマークは，花の中に象が描いてあるのですが（当時，このマークは象さんの花マルキと呼ばれていました），花マルキは間違いなしとワクワクしていました。

が，花マルキがもらえたのは，小鳥や花，電車などを作った子たちでした。

子供にとっては，ウルトラマン一家も怪獣も現実に存在するものです。大人がデザインした作り物ということになると子供の夢を壊すのでは，という制作側の配慮から，怪獣はすべて実在のものとされていました。図鑑が発売され，そこには怪獣たちの身長，体重，食べるものや出身地などが克明に記されていました。それを見てますます，もっと上手に怪獣を作りたい！　そして花マルキをもらいたい！　という気持ちが高じるのでした。怪獣好きの同級生たちと，作る種類が重ならないようにあらかじめ打ち合わせをしたり，図鑑の交換などをしてお互いに情報の共有化と充実を図り，設計図などと称するものまで描いて制作にはげんだ結果，油粘土作品は，初期のころに比べるとはるかによくなっていきました。少なくとも，作っている本人たちにはそう見えたのです。

油粘土は柔らかいので，２本足で立ったポーズを作ってもくずれてしまいます。解決策として，中に針金を入れる工夫をしました。その点，CGの仮想空間では，重力を気にせずにモデルを作ることができますけれどね。

こんなふうにして，図工の時間のたびにせっせと，ウルトラマンシリーズの怪獣たちを同級生とこしらえていましたが，やはり先生の反応はいま

ひとつ。怪獣を作った子に，象さんの花マルキがつくことはありませんでした。

一生懸命作っているのに，なんで自分たちだけ花マルキがもらえないのよ〜と，子供心に納得ができないのでした。

聞くところによると，怪獣は現代美術の若手作家がデザインしたものもあったということです。そういえば，ウルトラマン一家の着ているスーツの模様は，どことなく岡本太郎的なところがあります。子供向けのものといっても，けっこう美的ではあったのです。では，なぜ花マルキがもらえなかったのか……。

いまにして思えば，先生は，戦いをモチーフにしたテレビの怪獣などよりも，もっと穏やかで平和なものを子供に作ってほしかったのかもしれません。

今回のCGデータ

1280×1024ピクセル
1670万色フルカラーを 4 × 5 ポジで出力
使用ソフトはC-TRACE
総物体数104(物体数71, メタボール数12, 論理演算21)
点光源 1
マッピングデータは，スキャナCZ-8NS1による取り込み画像。「響子in CGわ〜るど」のCGは，ピクセルの縦横比（アスペクト比）1：1でレンダリングしたものを，ポジ出力しています

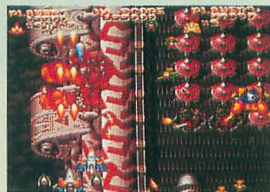
SOFTWARE INFORMATION

格闘ゲームラッシュが過ぎると次はシューティングゲームの番かな。今回紹介した2本もゲームセンターでなかなか人気のあったもの。発売が待ち遠しいね。あと，突然発売が決まった「上海Ⅳ」も見逃せないぞ。

魔法大作戦

「魔法大作戦」は，ファンタジックな世界を背景にした，硬派な縦スクロールシューティングゲームです。操作するキャラクターは性格の違う4人。強腕戦士ガイン，魔法使いチッタ，侍竜ミヤモト，呪術師ボーンナムです。彼らの目的は敵のボスであるゴブリガン王にかけられた賞金。2人同時プレイもできます。

ステージは全部で7面。

攻撃方法は，アイテムを取るとパワーアップをするショットとピンチのときに使う有限ボムなので，比較的ポピュラーな形式の縦スクロールシューティングゲームです。難易度はやや低めなので，縦スクロールシューティングゲームの好きな人なら，初めてのプレイでも，たぶん5面ぐらいまではいけるでしょう。個人的には，最初の面の弾のスピードは，妙に遅い気がしたんですが……。

アーケード版の完全移植とのことですが，手元の開発バージョンでは，詳細は不明。（瀧）

X68000用　5″2HD版　9,800円(税別)
EAビクター　☎03(5410)3111
画面は開発中のものです

今度はどれで遊ぼうか

	(前回順位)
1．スーパーストリートファイターⅡ	—
2．プリンセスメーカー	—
3．スターラスター	—
4．魔法大作戦	7
5．スタークルーザーⅡ	3
6．サムライスピリッツ	4
7．XDTP SX-68K	2
8．VIEW POINT	—
9．ディグダグ/ディグダグ2	—
10．クイーン・オブ・デュエリスト外伝α＋	—

9月号の読者アンケートはがきのなかの「期待の新作ソフト」に挙げられていたものを集計しました。先月号は集計結果が掲載されませんでしたので，前回順位は8月号のものです。2カ月の間があいたせいか，今回初登場のものがたくさんありますね。

特に，1位の「スーパーストリートファイターⅡ」はいうまでもなくダントツの支持率です。今月号のレビューは評価版によるものですが，製品が店頭に並ぶのは9月末。もうすぐです。

他機種の発売からはかなり遅れたものの，やはり期待の声の多かった「プリンセスメーカー」が2位。発売は来春です。

3位の「スターラスター」は電波新聞社から発売。アンソロジーシリーズではありませんが，やはり懐かしのシューティングゲームで，すでに発売ずみです。詳しくは28ページを。

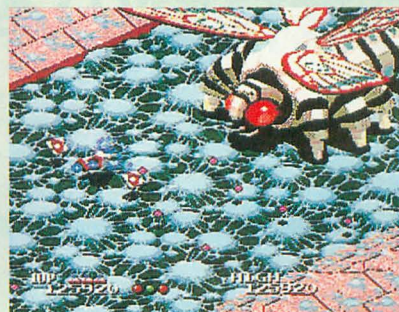
4位の「魔法大作戦」は発売予定がやや延びて12月となりました。華やかな画面は上の写真でもわかるとおりです。ファンの方はもう少しお待ちくださいね。

VIEW POINT

クォータービューの疑似3Dシューティングゲーム「VIEW POINT」。11月の発売に向けて開発進行中である。移植作品だが，X68000版ではパソコンならではの機能もいくつか追加される予定。

アーケード版にはないコンフィグ画面では，4段階の難易度を選択できるようになっている。ハイレベルモードはアーケード版を凌駕する「ウルトラシューティングマニア向け」とのことなので，腕に覚えのある人はお楽しみに。

そのほか，パッドの移動方向制御の設定ができる「PAD MODE」機能や，MUSICテスト，SOUNDテストの機能も搭載しているという。

6つのステージは，それぞれ「都市」「遺跡」「森」「山岳」「溶岩」「宇宙空間」のイメージで構成されている。

X68000用 5″2HD版 7,800円(税別)
ネクサス インターラクト ☎03(5474)3581
画面は開発中のものです

レッスルエンジェルスSPECIAL

TAKERUから「レッスルエンジェルスSPECIAL」の発売が決定した。

今回のゲームモードは3つ。そのなかのワールドサーキットモードでは，純粋にひとりの選手を育てていく。最初は弱小団体の新人としてデビューする。あとは試合を重ね経験を蓄え，世界王者を目指すのだ。さまざまな出来事が起こり，その選択によってシナリオが変わっていくぞ。

前作ではなくなっていた水着剝ぎデスマッチが復活している。ただし，18禁なのでパッケージのみの販売。発売開始は11月の予定。

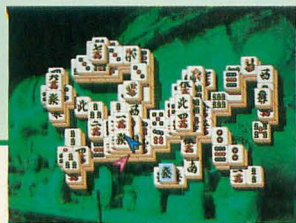
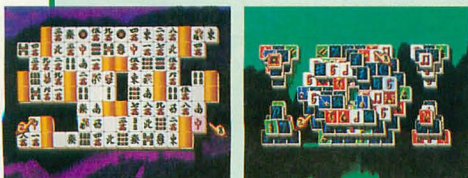
X68000用 3.5/5″2HD版 価格未定
TAKERU ☎052(824)2493
画面はPC-98版です

上海Ⅳ 万里の長城

人気のパズルゲーム「上海」の最新作が，11月にパソコン・家庭用ゲーム機の各機種で同時発売される。

この「万里の長城」にはいわゆる普通の「上海」のほかにも，ゲームモードがいくつかある。牌が横向き配置で，下から取ると上の牌が落下する「グレートウォール」，隙間を作れば牌を動かすことができる「北京」など，ひとひねりしたルールで楽しめるようになっている。

2人同時プレイも可能。

X68000用 5″2HD版 8,800円(税別)
EAビクター ☎03(5410)3111
画面は開発中のものです

発売中のソフト

★スターラスター 電波新聞社 8/26
X68000用 5″2HD版 5,900円(税別)
★F-Card V5 for x68k クレスト 9/1
X68000用 5″2HD版 12,800円(税別)
TAKERU 3.5/5″2HD版 12,800円(税込)

新作情報

★スーパーストリートファイターⅡ カプコン 9/30
X68000用 5″2HD版 9,800円(税別)
★VIEW POINT ネクサス インターラクト 11/未
X68000用 5″2HD版 7,800円(税別)
★魔法大作戦 EAビクター 12/未
X68000用 5″2HD版 9,800円(税別)
★X CASE Béシステム
X68000用 5″2HD版 19,800円(税込)
★ロボスポーツ イマジニア
X68000用 5″2HD版 価格未定
★Traüm 象スタジオ
X68000用 5″2HD版 価格未定
★鮫！ 鮫！ 鮫！ KANEKO
X68000用 5″2HD版 価格未定
★達人 KANEKO
X68000用 5″2HD版 価格未定
★エアバスター KANEKO
X68000用 5″2HD版 価格未定
★サバッシュⅡ ポプコムソフト/グローディア
X68000用 5″2HD版 価格未定
★麻雀悟空・天竺への道 シャノアール
X68000用 5″2HD版 9,800円(税別)
★スタークルーザーⅡ アルシスソフトウェア
X68000用 5″2HD版 価格未定
★地球防衛MIRACLE FORCE カスタム
X68000用 5″2HD版 価格未定
★XDTP SX-68K シャープ
X68000用 3.5/5″2HD版 価格未定
★プリンセスメーカー ニュー
X68000用 5″2HD版 14,800円(税別)
★ディグダグ/ディグダグ2 電波新聞社
X68000用 5″2HD版 価格未定
★レッスルエンジェルスSPECIAL TAKERU 11/未
X68000用 3.5/5″2HD版 価格未定
★上海Ⅳ 万里の長城 EAビクター 11/未
X68000用 5″2HD版 8,800円(税別)

対戦格闘の本流を見よ

Nakano Shuichi
中野 修一

いよいよ注目の大作の登場です。完成度も高い分難易度も高めですので，ゲームクリアを目指す人向けに鍛え抜かれた姑息な技を紹介しましょう。ただし，対戦では通用しないのでそのつもりで……。

いまやゲーセンを見てもコンシューマを見ても，対戦格闘ゲームなど掃いて捨てるほど転がっているのだが，そのなかでもストIIシリーズは別格の感が強い。瞬間的な話題性では新作スパンの短いNEO・GEOの作品に負けることも多いのだが，最終的な評価ではまだ並ぶものはない。現時点で考えても最初のストIIが素晴らしい完成度だったのに比べればバーチャファイターなどもまだまだ大味に思える。

ストIIの影響でどこもかしこも対戦格闘に走ってしまい，おかげでゲーセンの品揃えがつまらなくなったのも事実だが，いまだにストIIを超えるものがないというのはさらに不幸なことなのかもしれない。

もう何年もやり込まれたゲームだけに，突飛な技など出てくるはずがないのだが，極められ手詰りになったと思われていた対戦ダイアグラムが突然更新されていったり，有限の組み合わせのなかで次々に新しい技が開発される。

ゲーム自体の力というよりプレイヤーの凄まじさというべきだろうが，それもひとえにソフトにそれだけの魅力と懐の深さがあったということだろう。

ともあれ，X68000版スーパーストIIがまもなく発売される。スーファミ版などが先行発売されているのは衆知のとおりだが，原作の美麗なグラフィックが描き直されているのはちょっとつらいものがある。

X68000用　5"2HD7枚組　9,800円(税別)
カプコン

キャラ選択。色にもこだわろう

X68000版の移植度は，サンプルを見る限りキャミィステージでのオーロラの色変化が再現されていないという程度で，ビジュアルな部分は非常に忠実である（この部分で心配している人などいないとは思うが）。

サウンドは，現状では内蔵音源版BGMにストIIダッシュのときのものが流用されている。懸念のあった効果音は改善されているようだ。変にエフェクタを通したようだった音質のAD PCM音もちゃんとクリアになっている。

もうひとつの懸念であったMIDI版の音楽だが，SC-55で聞く限り，かなりまともなものになっているといっていいだろう。妙に豪華なシンフォニックサウンドがゲームとあっているか？　というと自信がないが，未完成な現時点で見ても全体的にデキはいい。前作での問題点に対して改善の姿勢が見られるのは高く評価したい。

操作感覚などを細かく比べてみても，現段階のものと本物を比べて違うところといえば，せいぜい百裂キックの持続時間程度しか見つかっていない。それさえ発売版では修正されていることもありうる。全体的に非常に忠実な移植である。「ボタンを押すタイミングがずれるとこうなりがち」といった部分も再現されている。本物を横に置き，CPSファイターではナンなのでセイミツの6ボタンコンパネをつないで試しているのだが，連射時のボタン受け付けがイマイチかなという程度で（並べてやらないとまずわからないと思う），まったく違和感はない。操作感覚はまさに本物そのものだ。

なお，アーケード版にいた3人目の女性キャラ（？）は性転換されてしまったようだ。

ただ，残念なのはランク設定が基板のディップスイッチで設定されるものと異なっていることだ。これはダッシュでも再現されていなかった。もっとも，本物もロットによってかなり差があるらしいのではあるが（編集室にある基板はどう考えても巷のゲーセンのものより難しい）。

難易度がどう違うのかというと，完全に甘口である。例によってもっともランクを上げた状態でプレイするのがよさそうだ。

はっきりいって，アーケード版のスーパーストIIのCPUランク設定が適当であるとはまったく思わない。個人的には，ちゃんとした難易度ならもっともっとヒットしただろうと思っている。キャラにもよるが，HARDESTなんか基板を叩き割りたくなるくらいキツい。EASIESTでやっと普通くらいだ。

従来のストIIシリーズではランクが上がるというのはCPUのアルゴリズムが変わってくるということだったのだが，スーパーではさらにダメージによるハンデキャップをつけている。ランクが上がっていると，同じキャラで同じ技同士の相殺があったとしてもプレイヤーが受けるダメージは圧倒的に大きくなるのだ。

現時点ではオーロラの色が違っている

（画面はすべて開発中のものです）

確かにこれがそのまま移植されてもシャレにならないかもしれない。
しかし，X68000版で鍛えてもゲーセンにいけば痛い目にあうのは目に見えている。これがマイナーゲームならバランスを取り直して移植するのもわかるが，ストIIシリーズは大ヒットしたメジャー作品なのだから鬼ゲーは鬼ゲーのまま移植するのが正しい，と私は思う。

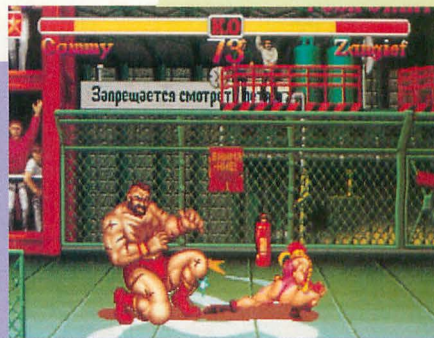
寄ってきたらいきなり蹴ってみる

足先だけ当てる間合を保つ

10MHzで実行するとどうなるのか

移植度以上に多くの方が気にしているのは動作環境に16MHz以上が推奨されていることであろう。
さっそく10MHzで試してみよう。
確かに重い。
スプライトの書き換えが多く発生していることが原因のようだ。新キャラ以外のステージではストIIダッシュと大差ないはずなのだが，スーパーのほうが重めになる。ノーマルキャラもパターンが増えていることが影響しているのだろうか。
特に重いのは背景で動いているキャラの書き換えだ。たとえば新キャラでもフェイロンのステージではゲーム中，背景はなにも動かないようになっているのだが，この状態だと10MHzでも実に軽快に動く。はっきりいって本物より速い。Xのターボ2くらいの速度だ。
全体的に書き換えの多い動作，たとえばジャンプ中に身体が回転したりするとぐぐっと重くなるという感じになっている。
これでも遊べなくはないが，16MHz以上ならまったく遅くならないので「16MHz以上推奨」というのは間違いではないだろう。さらに68030では前作同様に音声多重でプレイできる。各ステージ間のデータ展開速度も速いのでまったく快適だ。

対C戦の攻略

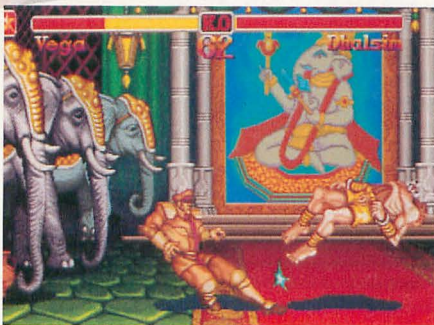
ゲーセンでは対戦基本のゲームだが，家庭ではなかなかそうもいかない。さっそく対C戦の攻略に入ろう。
……困った。なにせ選択できるキャラが16人もいるのだ。アーケード版では私はまだ11人（基本8人＋サガット，バイソン，キャミィ）でしかクリアしていない。その分若干片寄った攻略になるのでご了承を。
さて，アーケード版より甘いとはいうものの，やってりゃランクはどんどん上がっていくゲームなので簡単にクリアさせてくれないことには変わりはない。「どんな相手にも苦手な攻められ方がある」なら，それを知ることが必要だ。

なお，全体的に，昇龍拳（系の技）で小を見せ技に使い着地で大を出す，というパターンがよく効くので念頭に置いておいてほしい。

基本キャラ攻略

●対リュウ
足払いキャンセルを多用してくるようになったが，基本的なアルゴリズムはダッシュと変わらない。コマンド系の飛び道具があればそれだけで押せる。なければ波動拳を跳び込んでいく。波動拳を出す動作と同時に跳んで叩く。間合が近くなりすぎると食らい投げされるので深くなりすぎたときはなにもせずに，着地と同時に投げる。波動拳を完全に先読みして跳び込むと昇龍拳で返されるので，タイミングはほどほどに。
ダルシムはスライディングとドリルで，ベガはデビルリバースを交ぜて戦うことになる。ザンギエフなら転ばせたあとを追いかけて積極的に大足払いを重ねてみよう。

●対ケン
ランクが上がると稀に着地昇龍拳を決めてくるので，投げるなら空中投げが望ましい。

●対ダルシム
春麗なら踏むだけ。
テレポートの関係で基本的な動きがかなり変わっているが，まあこれは慣れるしかない。

●対本田

スライディング，3歩下がってまたスライディング

ダッシュに比べるとかなりいやらしくなっている。飛び道具がなく跳び込みの弱いキャラにとっては鬼門か。スーパー頭突きを返せるか否か，張り手を止められるか否かも重要。
跳び込みは遠い間合から相手の頭よりかなり前を叩くつもりで行うこと。ケン，キャミィなどはどんどん跳び込んでいこう。飛行軌道の長いベガなどは間合に注意。跳び込み攻撃の判定の弱いキャラではさらに慎重に。

●対ザンギエフ
キャラによってはかなり強敵。飛び道具があればかなり楽。飛び道具のないキャラでも，バイソンなら大パンチのみ，本田なら頭突きのみ，キャミィなら爪先をかすらせるような中足払いのみ。ブランカは跳び中パンチ＋頭突きの3段で押せる。跳び大キック＋中足払い＋ローリングの3段を狙うのも手。ザンギはしゃがみ大パンチ。

●対ガイル
3段攻撃を決めたり，ソニックを人間っぽく撃ったりと多少変化はあるものの基本的にダッシュと変わらず。攻略も同じ。春麗の中気功拳はよく食らってくれるので多用すべし。ブランカのバックステップの使いどころはここしかない。
リュウ/ケンの大パンチはちょっと弱くなっているので遠めの迎撃には使わないほうがよいかも。

●対春麗
驚異的にパワーアップしている。飛び道具と待ちというガイルパターンだが，踏みと削りがある分だけいやらしい。
小気功拳が当たらないキャラなら待ちに入ってもいいが，百裂で削りにくるので厄介だ。跳び込んでいく場合は迎撃の大キックを一方的に沈められる間合をつかむこと。
バイソンはヘッドバットが決め手。うまくすればなぜか20回くらい入る（そのほかキャミィ，ブランカ，バルログには非常に有効）。

無防備な落ち際は狙い討ちだ

パンチで落としてさらに着地点を狙え

ザンギエフは気功拳で削られる分は無視して寄ってくるのを待つ。三角跳びは後ろから大キックで落としたあと，連続でジャンプ大キックを何回か重ねてみよう。近寄ってくるときにはストマッククローを狙うのも手だ。

ダルシムの場合は,「寄ってくる相手への中キック」というパターンが一部通用する。

●対ブランカ

ダッシュでは初心者ストッパー的な役割をしていたCPUブランカだが，スーパーでは弱体化し隙も多くなっているので比較的楽な相手。無敵技がないので，転ばせたら起き上がりのところに大パンチ（ブランカなら大キック）をめり込ませるようなタイミングで跳び込んでみよう。

新キャラの攻略

新キャラについては，これまでのキャラとは基本原則が異なると考えたほうがよい。「対空は無敵技でやるもの」というのがスーパーの基本思想のようだ。

●対ホーク

最大の脅威は１回転の大足払いである。メキシカンタイフーンなら２発くらい食らってもどうってことはないが，ランクが上がっていると足払い３発であの世行きである。まずはこれをしっかり見切ること。

次にコンドルダイブ対策。

本田なら頭突きを出すだけ。

春麗ならスピニングバードで返す（中がいいかな？)。ザンギならラリアットかタイミングよく大キックを出す。

ブランカほか突進系の必殺技があるキャラはとりあえずガードして，相手の着地前に落とす。

リュウ/ケンは立ち大パンチが無難。どうしてもパーフェクトを取りたければ，最初になにか攻撃を当てて中間間合でひたすら立ち小キック連射（つまんねー)。

ダルシムは，先読みできたときは早めに中足を出すことでダイブを押さえることができる。対戦で使えば最高の屈辱技だ（ただし失敗すると痛い)。普通にやるならタイミングよく立ち小パンチだ。迎撃が間にあわなかったときはガードしてすぐさま斜めジャンプ中キックなどで返す。

ガイルは（サマーソルトで落とせばそれまでだが)，ラウンドハウスキックで落とすのが美しい。早めに足を振り出して，爪先でピシッと止める感じだ。

＊　　＊　　＊

基板買った人がかわいそーじゃないですか

このゲーム,「要4Mバイト，16MHz以上推奨」という思い切った設定で発売されるわけだが，このソフトの売れ行きに注目する向きも多いだろう。噂ではX68030専用ゲームを作成しているソフトハウスもあるというし，HARPもそろそろ出荷されているかもしれないし……。

しかし，さすがに凄い移植だ。

10MHzだと重くはなるが，初代のX68000でも最新のゲームがそのまま動くというのは考えてみると凄いことかもしれない。１年前にはまともにストIIが動くかが話題になっていたのに，いまや新鋭CPシステム２上のスパIIが実際にそのまま動いてしまうのだ。

製品版では，一時省略されるといわれていたトーナメントモードが復活する。ゲーセンでもあまり置いてなかったのだが，結構面白いシステムだったと思う。まあ，４台で通信しないと意味がないような気がするが，個人的にはうれしい機能だ（８人台がなくなったので優勝したことはまだないのだ)。

これでストIIシリーズの２つが移植されたわけだが，残りのストII，ターボ，Xに関してもサポートがほしいところ。このシリーズはどれも捨てがたいものがあるのだ。さらにいえば，Xの本田でIIのガイルと対戦できたりすれば完璧なのだが……。

総合評価

	0　　5　　10
グラフィック	★★★★★★★★★
音楽	★★★★★★★★
操作性	★★★★★★★★
忠実度	★★★★★★★★★

余裕があれば垂直ジャンプでかわしてそのまま連続攻撃にもっていく手もあるが，リスクが大きく，反応が速いキャラでないとちょっと無理。

●対ディージェイ

もっともてこずるのがこいつ。

昇龍拳を持っているキャラでないと完全な対空は不可能。スライディングに隙がなく，下手するとエアスラッシャーの連発にコマンド技の波動拳が打ち負ける。

対策だが，ガイルならなにも問題なし。

リュウ/ケンは波動拳で押せる。近寄ってきたときは牽制して跳ばせたいところだが，できるだけ足を出さず，出すとしても小足で牽制すること。

本田は頭突きで対空。小頭突きより大頭突きのほうがよさそうだ。地上は中張り手で牽制。エアスラッシャーをちゃんと避ければ，パワーで押せる（かな？)。

春麗なら空中戦でまだ優位に立てる。基本は遠距離から中キックで跳び込んで，足払い，すかさず踏みにいき踏みまくる。踏んだあと蹴り落とされるようなら空中スピニングで逃げる(理想論)。地上では中足払いをガードさせて小スピニングというのを連続で出すとかなり当たってくれる。

ブランカは基本性能で圧倒的に不利なので待ちながらチャンスを狙う。中足払いを基本として，対空はバーチカルローリングで。それでもたまに負けるが，悪くても相打ちにできることが多い。それ以外の対空技はほぼ無力。逃げ昇り大キックなども一方的に負ける。相手が倒れたらジャンプ大キックで跳び込む。攻め込むなら跳び中パンチ＋頭突きが有効。

ザンギエフは近寄ってきたらストマッククロー，余裕があれば中パンチでずんずん投げにいく（投げハメのように）と有効なことも。

ダルシムはややつらい。遠めならファイヤーで牽制し跳んできたら着地点に立ち大パンチ(ほかの対空は×)。近めで当てたときは２回連続で大パンチを押してみよう。跳ね飛ばした先でもう一度当てられることがある。頭上（直上）にきたらチョップで落としてファイヤーで削る。直上よりも微妙にはずれているときは思い切って中キック，または単に防御のみ。ヨガスナイパーカスタムは通用しない。

●対キャミィ

基本アルゴリズムは単純。体力が半分になるまでは跳び込んできて，それ以降はスパイラルアローだ（ただしランクが上がると少し変化するので注意)。よっていかに体

力を半分にするかが決め手となる。
前半の跳び込みは強力なので受けに回りがちだが，春麗ならキャノンスパイクを踏み潰せるので積極的に攻める手もある。
ザンギエフはスクリューなどよりもストマッククローを狙うのが基本だ。春麗では鳩尾(みぞおち)に決まってイマイチだったが，キャミィは背が低いので……まあそういうことだ。

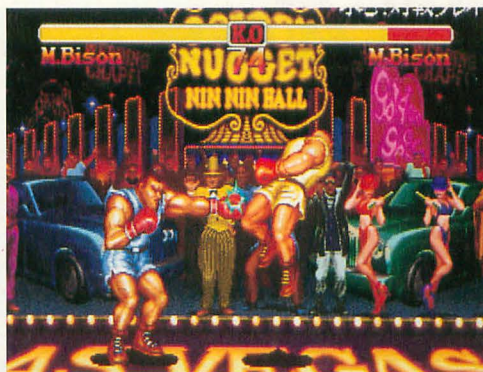
●対フェイロン

やけに堅い。十分な対空兵器がないとちょっとつらいかも。ガイル，ブランカなどの遠距離対空のしっかりしたキャラでは特に問題はないだろう。投げキャラなら，ランクが低いうちは投げハメっぽく攻めきるのが手っとり早い。
リュウ/ケン使いで体力に自信がある人は大道芸として，小足払い連射で牽制し，跳んできたら立ち小キックで落とすという手がある(連射装置は邪道)。ただし確実性はやや薄い。近寄ってのしゃがみ掌底はしゃがみ中パンチで返せる。
ザンギエフは楽勝。寄ってきたらタイミングよく中パンチでストマッククローを決める。防御体勢のまま出せるうえ，実によく吸い込む。
歩いてきての2段蹴りは，転ばせるチャンスだが，キャミィではタイミングが悪いとまともに食らうので注意。
攻め込んでくると最後に掌底を当てて跳び退いていくが，ここでローリングなどを当てることもできる。ただしダッシュストレートは半拍置いて出さないとうまく当たらない。

クリア目前四天王戦

通常キャラの7,8人目がきつく，新キャラがやっかいなのに比べれば四天王は比較的楽に倒せる。バイソンが出てくるとほっと一息つける感じ。

●対バイソン

たいした問題はないだろう。ダッシュアッパーを転ばせるのが穏当だが，飛び道具があれば（中がよい）タイミングよく連射するだけでもいい。
春麗は中足払い，踏みつけのループが有効。うまくすれば百裂キックのみでパーフェクトも(ただし不確実)。本田なら百裂で前後しながら牽制するだけ。
大味なのがバイソンvsバイソン。ダッシュ攻撃はすべて早めのストレートで返せる。おすすめは，キックを溜めておき，ひたすら小パンチ連射(レバーは使用しない)。これでほとんどの攻撃を返せる（確実ではないので注意)。ちびちび削ったら，あとは40秒近辺でタイミングよくファイナルを決める。

●対バルログ

中距離で出足に中足払いを置いておくとガンガン削れる。転ばせたら跳びかかって，バック転してくれたら追いかけて投げる。
春麗なら開始直後にあとずさる相手をすかさず追い詰め，小キック跳び込み。ジャンプで逃げようとする相手を落として着地と同時に百裂で削る。あとはこれを繰り返してみる（ランクが上がっていると返されることがある)。
ダルシムの中スラパターンは目押しになったが結構使える。

●対サガット

だいたいダッシュと同じ。まあ無謀なことをしなければ大丈夫だとは思うが……。

●対ベガ

空中投げできるキャラは確実に決めること。飛び道具や突進技は，転ばせて起き上がった直後に一瞬置いた感じで当たるように調整する。
サイコを潰せるキャラならかなり楽。逃げ大キックとかジャンプ大パンチとか……。自分の使うキャラでなにが有効かを各自で探してみよう。
サイコは出かかりが弱いので，リュウ，ケン，春麗，ブランカなどは転ばせたら大パンチで跳び込むとよい。
あい変わらず楽なのはブランカ。ダッシュのとき同様，ローリングだけで勝てる。足払いも一方的に勝てる。
春麗も中足払いさえ当てれば，深めにガードさせてジャンプ中キック，踏みつけのシーケンスや中小スピニングからの投げが簡単に決まる。
本田はブランカとほぼ同様で大丈夫。
まともにやるとまず勝てないのがダルシム。ダッシュのときもかなり苦しかったが，今回はそれ以上だ。適当な間合で昇り大キックを出し，着地投げをするしかなくなった。

最後に

一貫してブランカ使いだった私は，スーパーになってからのあまりの弱体化に悩まされた。制空権は奪われ，性能アップもなく，攻撃判定は縮小され，必殺技はいとも簡単に落とされ，キャンセルも効かず，増えたのは使えない技だけ……。
一時は春麗（白）に転んだこともある。
やがて気を取り直してブランカ（オレンジ）に戻ったのだが，すでに巷ではブランカ使いをみかけなくなっていた。Xでほんの少し復権したものの，あい変わらずさびしい状況だ。そういえば今年のゴールデンウィークの最大のトピックスは「ジミー君ゴウキを倒す」だったっけ……。
メインキャラはブランカに落ち着いたものの，ほかのキャラも使えば面白い。コンボゲージが出るようになって以来，ケン(赤)のアッパー昇龍拳も快感だし，最近の対戦ではめっきりダルシム使い（黒）になっている。クセが強いので敬遠している人も多いかもしれないが，ダルシムって使ってて面白いキャラだと思う。
せっかく家庭でこれだけのゲームができるのだから，ぜひ全キャラマスターを目指してみてほしい。

残り43秒まで耐えてファイナルを決めろ

隙があればどんどん空中投げだ

最後の頭突きはほぼ決まる（ガードされても）

甦る星々のきらめき

Yaegaki Nachi
八重垣 那智

ビデオゲームアンソロジーシリーズはちょっとお休みで，電波新聞社の新作はファミコンで人気を博したシューティングゲーム。X68000用にアレンジされての登場です。宇宙空間が舞台の戦いを，腕と戦略で勝ち抜きましょう。

いまを去ること十数年前，パーソナルコンピュータなる言葉が生まれるよりも少し前に，マイコンで流行したゲームのことを覚えている人はいるだろうか。自作のマイコン(CPUは8080！)に数KバイトのBASICインタプリタを載せ，1Kバイトや2Kバイトのフリーエリアにソフトを書いていたその時代のヒットゲームといえば，「スタートレック（宇宙大作戦）ゲーム」にほかならない。

表示はすべて文字表示。入力は対話型であるが，コマンドはすべて数値入力。マップを毎回画面に呼び出しながら，ゲームを楽しみ，さらに自分でソフトを改造してゆく。それは，そんな時代だからこそ愛され楽しむことのできた，マイコンならではのゲームであった。

そこに宇宙があるから

ナムコから1985年にファミリーコンピュータ向けにリリースされたオリジナルゲーム「スターラスター」が，このほどナムコと電波新聞社によりX68000上に移植の運びとなった。このゲームは冒頭で述べた，マイコンで大流行したスタートレックゲームがファミリーコンピュータに移植されたかのような画期的なソフトであった。一見まったくの別ゲームだが，その根底に流れている戦略性を重視するコンセプトには大きな違いはない。少なくとも，同じ面白さを再現するためにスターラスターは生み出されたといっても，きっと間違いではないだろう。

命中すれば快感

発売されてから9年の歳月がたち，ファミリーコンピュータがすでに過去のゲーム機となっても，いまだに名作としてこのゲームの名を挙げる人がいるという事実が，その確かな内容と印象のよさを示している。当時，ゲームセンターでしかゲームを遊ばなかった私は，ナムコ直営店に置かれたゲームセンターバージョンで，このソフトに触れたのだが，あの勇ましいファンファーレを聴けば，いまでも当時の記憶と共に目の前に宇宙空間が広がるのである。

戦いは宇宙を染めて

このゲームにおける目標は，あくまでも敵を全滅させることである。全体マップ上でEの文字である敵に向かってワープし，そこでリアルタイムの戦闘を行う。戦闘中，レーダーはコンバットモードに切り替わって敵の位置と距離を表示するので，うまく接近して攻撃を加えなくてはならない。数機の戦闘機とボール状の母船を全滅させると，マップからEの文字も消滅する。これを繰り返し，敵が全滅するまで戦闘は続くのである。

敵の攻撃や戦闘で消耗したエネルギーが0になるとゲームオーバーになるので，そうなる前にマップ上のBのエリアにいる基地にドッキングして，エネルギーの補給を受けなくてはならない。ただし，これらの戦闘経過は随時チェックされており，ゲーム終了後にプレイヤーが評価されるときには，過度の補給や時間のかけ過ぎはマイナス要因となる。できれば補給も控え目が望ましい。よい評価を得るためには，とにかくムリ・ムダ・ムラをなくすことが重要だ。

ゲームはスタート時にTRAINING，COMMAND，ADVENTUREの3つのモードから選ぶようになっている。いま述べた共通ルールについてはすべて同じだが，もちろん各モードには，その難易度と共にいくつかの違いがある。プレイヤーの腕によって遊び分けるのではなく，あくまでもプレイヤーが上達した証として，より難しいモードに進むという考え方のほうが正しいだろう。

よって，やさしいモードというのは上達のためのレクチャー的な要素を含んでいる。まず，TRAININGモードは，マップ上には基地と敵だけで，その名のとおりの主に戦闘の技術を学ぶモードである。レーダーの見方や，敵を正面にもってくるといった基礎はここでマスターしておきたい。

守れ，そして考えよ！

戦闘技術が身についたところで，COMMANDモードにレベルアップすると，マップが急に賑やかになっていることに気づくだろう。マップ上に登場した＊マークは惑星であり，プレイヤーが敵の侵略から守り抜かねばならないものである。ここでプレイ

もたもたしてると基地が危ないぞ

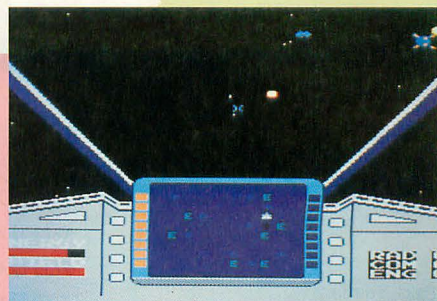
X68000用　5″2HD版　5,900円(税別)
電波新聞社　☎03(3445)6111

ヤーが鍛えるのは，より危険度の高い敵を優先的に叩くという判断力である。ここに，このゲーム独自の戦略性が生まれ，プレイヤーに戦闘以外での思考力を要求してくることになる。より厳しい戦いのために，基地からは補給以外にパワーアップを受け取ったり，光子魚雷(個数限定)でマップ上の敵を攻撃したりすることも可能になるが，しょせん頼れるのは己の腕だけに過ぎないことをすぐに思い知らされることになるだろう。

そして最後に挑むのはADVENTUREモードである。惑星や基地を守るだけではなく，ビッグバンを引き起こそうと企む敵の本拠地「暗黒惑星」を突き止め，敵の野望を阻止しなければならない。そのためには惑星からキーワードを受け取って，これを解析しなくてはならない。より複雑さが増しながらも，ここにスターラスターの真骨頂が現れる。

敵との戦闘を続けながらキーワードを集めに惑星を巡り，パワーアップを求めて基地も廻る。そして最後にキーワードを特定の惑星で解析するという苦行があってこそのスターラスターなのである。作戦を成し遂げたときの感激には，また格別なものがあることは間違いないだろう。

名作という見えない枠

今回発売されたX68000版は，あくまでもオリジナル版を重視してあり，アレンジモードといえど，グラフィックとサウンド以外に目立った変更点を見出せない。アナログジョイスティックに対応するとか，ときおり隠れキャラ(真のボス？)が出るといった，基本的には戦闘モードに関した地味な変更のみであり，X68000ならではという要素がさびしいというのが正直なところだ。

確かにオリジナルに比べれば視界は広いし，グラフィックも綺麗だ。特にワープなどはオリジナルで迫力がなかったところだから，非常に目立つ違いだろう。レーダーが半透明なのも，アリガチなのだろうが嬉しくなる仕掛けだ。自動連射なんていうサービスもある。でも，やっていることは何も変わっていないのである。そういう意味ではオリジナルモードに戻しても，何も変わらずにプレイできてしまう。いま挙げた違いは，基本的にはゲーム自身を何もアレンジしてはいないのだ。

ここまで書いてきたように，スターラスター自体は，よいゲームとしての素質をかなり備えてはいる。しかしその半面，ファミリーコンピュータというハードウェアに制約されたり，過去の常識にとらわれたゆえの問題点も同時に抱えたゲームなのである。今回の忠実な移植で，そういったものも一緒になってX68000の上にやってきていることについては，やや残念に感じてしまう。

色数やキャラクター数の限界から，敵や惑星の個々の識別が全体マップでは不可能であることや，ゲームの難易度を大きく左右するにもかかわらず，ランダムに決定される，その初期配置(有利な配置が出るまでリセットを繰り返すのが，上級者の常識？)といったものは，より高度で本格的な戦略性のためには，改良されるべきであったと思わずにはいられない。

また，これ以外にも，感覚に依存した戦闘シーン(特に弾避け)にも，いまどきの解釈が欲しかったところだ。経験的にしか知り得ない敵キャラクターによる攻撃力の違いを視覚的に表現するなど，昔はそれでよかったものも，もう一度考えてほしかったと思うのである。それが9年前の名作に対する現在の回答ではないのだろうか？

マップを見ながらワープして，他エリアに移動

懐かしのオリジナル画面モードもある

基地でパワーアップだ

2つめのキーワード獲得！

星空を見上げて

X68000はすでに7年もの歴史をもったパソコンだが，ファミリーコンピュータにはできなかったことが，いっぱいできるはずである。だからこそ，進歩したゲームの概念と共に，新しいスターラスターの姿に期待したのである。ただ，これは私の思い込みで，そこにはちょっとおめかししたスターラスターが，昔のままの姿で立っていたのだった。その姿を見て頭に浮かんだのは，やはり懐かしさであった。

結局，これは名前こそ違えど，そしてアレンジ版が収録されているといっても，ビデオゲームアンソロジーのシリーズと，その性格は同じものなのであろう。スターラスターというゲームの素晴らしさをいまに伝えるため，余計な解釈は加えずにX68000ユーザーに贈られた，タイムカプセルだと考えるのがいいようだ。それを開けて何を思うかは，きっとそれぞれのプレイヤーにしかわからないことなのだろう。

過去の価値と現在の価値

見かけは3Dシューティングゲームなので勘違いしやすいのですが，その実は戦略性を重視したシミュレーションであるというのが個人的な印象です。そういった考えからいえば，そっち方面のグレードアップがやっぱり欲しかったです。また，演出もパワーアップさせて，基地から補給を受けたり惑星に接近したときには，もっとゴージャスなビジュアルシーンがあってもいいと思いました。

そういえば，昔から疑問なんですけど，惑星や基地から飛んでくるロボットって戻っていかないけど，使い捨てなんでしょうか？　だとしたら「HELLO」っていうだけってのは，すごくもったいないと思うな，うんうん。

総合評価（0　5　10）

項目	評価
ゲーム性	★★★★★★★
技術	★★★★★★
サウンド	★★★★★★★
グラフィック	★★★★★★★
やっぱり度	★★★★★★★★

お兄さん，いい娘がいまっせ

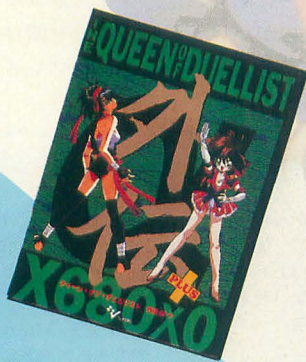
Taki Yasushi
瀧　康史

やはりゲームはキャラクター命，好みの娘じゃなきゃ気ものらない，そんなこだわりも生きていく上では必要ね。ここの登場の10人は，それぞれ漫画家たちによるデザインで，各々の個性が光ります。まずは一線，お楽しみください。

どうも最近，色モノっぽいゲームづいているみたいな私。おかげで，対戦ゲームをやっていてもどうしてもお遊びが欲しくなってしまい，スト2でも，もの足りなくなってしまったなぁ……。

今回紹介する「クイーン・オブ・デュエリスト外伝α＋」も，色モノ度たっぷりの対戦格闘ゲーム。どのあたりが色モノかというと，登場するキャラクターは女の子。しかも，勝つと相手の女の子は服をお脱ぎになるんだな。ただし，それは18禁のパッケージ版のほうだけで，TAKERU版は一般バージョンなので服は脱がない。ゲームそのものはまったく同じで，違いといえばパッケージ(箱)と脱衣の有無ぐらいかなぁ。

キャラクターデザインはすべて漫画家によるものだそうで，そのせいか，登場する女の子たちはそれぞれ違った表情を持っている。そして，多少の重複はあるようだが，個々のキャラはちゃんと声優が喋っているみたいである。

とにかくよく喋るゲームで，ハードディスクにインストールをして中身をみてみたら，なんとPCMファイルが3Mバイトにも達しているぐらいだから，どれほどか想像がつくよね。X68000のゲームでたぶんいちばんよく喋るゲームだと思うな。

X68000用
TAKERU　3.5/5"2HD版　5,800円(税込)
パッケージ(18禁)　5"2HD版　9,800円(税別)
TAKERU　☎052(824)2493

1Pは色も選べるぞ

10人の闘う人々

登場キャラクターは全部で10人。ほとんどが可愛い女の子だが，そうでない奴も若干混じっている。どのキャラがどんな漫画家によってデザインされたのか，声優は誰なのか，という点がある方面の方々にはとても重要なことだと思われるので，それも併せて，それぞれ簡単に紹介しよう。

Oh!FM TOWNS編集部の協力を得て，超必殺技，隠し技などがだいたいわかったのでプレイの参考にしてほしい。連続技の項は私がそのキャラでよく使うもの。「謎」とあるのは，うまい連続技がイマイチわからなかったキャラである。隠し技はいつでも使えるが，超必殺技は体力がレッドゾーンにならないと使えない。コマンドの表記は，自分が左側のときのものである。Pはパンチ，Kはキック，(PK)はパンチ＋キックの同時押しだ。技の説明はしないが，簡単なので自分で試してみよう。(タメ)というのは，その状態でしばらくレバーを保持することを意味している。

○鷹森ちづる(TAKAMORI CHIZURU)
キャラクターデザイン：蘭宮涼
声優：深雪さなえ
超必殺技：アイドル・トーピドー
←↓→(PK)
備考：いわゆるリュウケンキャラ

○舞野舞(MAINO MAI)
キャラクターデザイン：あずまきよひこ
声優：篠原恵美
超必殺技：クイーンスケアドロン
←↓↙(PK)
備考：ガイルにキム乱舞

○神戸麗華(KANBE REIKA)
キャラクターデザイン：海野やよい
声優：嶋村薫
超必殺技：舞舞(マイムマイム)
↓(タメ)←(PK)
備考：女王様

○光明寺キリコ(KOUMYOUJI KIRIKO)
キャラクターデザイン：マイケル原腸
声優：安藤ありさ
超必殺技：パワーコレダー
↓(タメ)→(PK)
備考：謎

○金城エルザ(KANAGUSUKU ELZA)
キャラクターデザイン：ここまひ
声優：白石文子
超必殺技：エルザフィニッシュ
(「エルザダンス」にしか聞こえないけど……)
←→↓P
隠し技　：空中投げ
お互いが空中にいるときに，レバー＋パンチ
備考：しっぽがちゃぁみぃ

○松田美由紀(MATSUDA MIYUKI)
キャラクターデザイン：
うたたねひろゆき
声優：小林優子

SHARPにキック！

超必殺技：黒龍乱舞　←↓↙(PK)
備考：リュウケンに乱舞つき

○**エミリー・バーネット**
(EMILY BARNET)
キャラクターデザイン：富士参號
声優　：篠原恵美
超必殺技：ネオライトニングクラッシュ
←(タメ)↓(PK)
備考：ガイルにカイザーウェイブ

○**龍凰**(RYU-HOU)
キャラクターデザイン：龍炎狼牙
声優　：豊島まさみ
超必殺技：火炎旋風陣　→↙↓↙←P
隠し技　：空中投げ
お互いが空中にいるときに，
レバー＋パンチ
2段蹴り
レバーを左右いずれかに入れて，(PK)
備考：どうみても小学生……

○**ストレンジ・ゴールド**
(STRANGE GOLD)
声優　：安藤ありさ
備考：謎

○**グレート・武神**(GREAT TAKEGAMI)
声優　：梁田清之
備考：唯一の男性キャラ

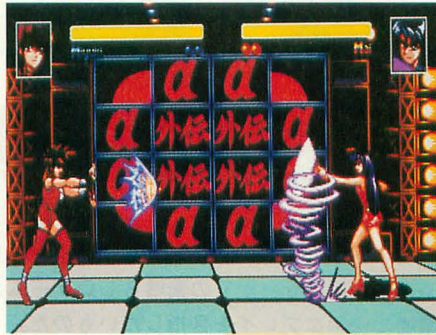
とまあ，こんなところである。私は漫画やアニメをあまり見ないので，声優や漫画家についてはよくわからないのだが……。

ほのぼのしながらラクラクゲーム

さて，X68000には格闘ゲームが各種発売されているが，このゲームは，ファミリーソフトの「あすか120%BURNING Fest.」のように実は妙にアツイ対戦ゲームというわけではないし，「ストリートファイターIIダッシュ」みたいに妙な駆け引きがあるわけでもない。ほのぼのしていて，タラタラとプレイしながら楽しめる。編集部でプレイしていたら，石上さんに「楽しそうですねぇ」といわれてしまったし。

スピードと難易度の切り替えができるので，自分の力量に合わせられるところもよい。連続技が入ると，キャラの名前の下にマークがつくので，何回キャンセル技が入ったか楽しめる。

どこが楽しいのか，と聞かれると，ひと言ではいえないが，とにかく面白いのだ。私がプレイしているところを読者のみなさんが見たら，いかに楽しんでいるかきっとわかってくれるに違いない。

理由のひとつには，「作業」となってしまうほど気合を入れなくても楽しめる，ってのもあるかもしれない。最近の対戦ゲームは，コンピュータが強すぎて，いわゆるアルゴリズムハメをしないと勝てないケースが多いけど，このゲームはごく普通にプレイしていればたいていは解ける。

この2人のどちらかが，たぶん最強なんだな

この技は自分の後頭部で攻撃する技だ(ウソ)

コマンド操作はほとんど一般の対戦ゲームと同じ。ただ，防御に上と下の区別がほとんどない。しゃがみキックはあっても，しゃがんでいる状態がない。したがって，防御は相手と反対方向と下方向全部で，ほかのゲームよりも簡単だ。

必殺技は，レバー制御＋ボタンだが，ボタンは押してさえいれば技が出るので，いっそのこと，レバーを入れたら押しっぱなしにしてもいい。超必殺技もあって，どれも強力なものだけど，コマンドは比較的ほかのゲームに比べては簡単なので(長くてヨガフレイム程度)簡単に出せるだろう。

このように，対戦ゲーム慣れしていなくてもプレイできるのは，非常にいいことだ。

ちゃんと，ちゃんとのアレンジだ

実は，X68000版をやる前に，PC-98版とFM TOWNS版をやってみた。PC-98版，FM TOWNS版，X68000版の順で発売され，そのたびに少しずつバージョンアップしている。ただ，FM TOWNS版とPC-98版の違いは，音楽リソースがCDになった程度で，グラフィックもお喋りもほとんどベタ移植であった(しかし，CD-ROM版なので音楽はFM TOWNS版がいちばんいい)。

そこへきてX68000版では，まずPCMが増えた。個人的にPC-98版とFM TOWNS版では，声優の選択が悪いんじゃないかと思ったキャラがひとりいたのだが，そのキャラのPCMがまるっきり変えられている。また，背景などのグラフィックの大半が新たに描き下ろされた。そのほか，キャラにスプライトを使っているので移動が滑らかになったとか，必殺技などのコマンドが簡単に入るようになったなどの違いがある。

ほかにも細かい違いはいろいろ。正直いって，久しぶりに，X68000向けに気合を入れてアレンジされてゲームを見たなった感じかな。

闘いは気の向くままに

最初見たときにはビックリした。というのも，ゲーム作りに妙に気合が入っているから。正直いって，やたらゲームを難しくしようと勘違いしているようなゲームデザイナーが多いなか，あえてゲームを簡単にするというのは，なかなかの処置だと思う。

とにかく気軽に楽しめるので，何かの息抜きなんかにもおススメ。これはぜひ一度やってみてほしいゲームだと思う。

戦闘環境はハードディスクにね

ハードディスクにもインストールでき，よっぽど変な環境でない限りは，たいていの環境から立ち上がる。プロテクトもないようなので，キーディスクはいらないみたいだ。ただ，RDNというPCM8アッパーコンパチのツールが入っているため，PCM8を常に常駐している人はPCM8を解除しないといけない。

PCMやグラフィックのデータが多いので，おそらく，ハードディスクやMOにインストールしなければゲームなんてやっていられないだろうけれど，これも時代の流れかな。ハードディスク対応というよりも，むしろそちらが前提で，フロッピーディスクにも対応といったところか。

残念なのは，ゲーム終了ができないこと。何かの作業の途中にでも，ちょっとゲームをして，そのまま戻ってきて作業を再開，ということができればよかったのに……。

それから，もう少し，当たり判定と質感の徹底，キャラクターパターンの追加を行ったら，もっともっと面白いゲームになったんじゃないかなぁ。

総合評価	0　5　10
お喋り度	★★★★★★★★★
グラフィック	★★★★★★★★★
キャラクター	★★★★★★★
音楽(曲)	★★★★★★★★★
音楽(データ)	★★★★
ラクラク撮影	★★★★★★★★★
お買い得度	★★★★★★★★

4つのボタンがパワーアップの第一歩

Nishikawa Zenji
西川 善司

発売されてすでに1カ月以上過ぎた。はまっている人も多いことだろう。今回は「飢餓伝説SPECIAL」の操作性を中心にレビューしていく。せっかくの素晴らしいゲームだから，快適な環境で快適なプレイを目指したいものだ。

ディスク9枚組，ハードディスクでは約13Mバイトもの容量を占有する超大作格闘アクションゲーム「餓狼伝説SPECIAL」(以下，餓狼SP)。今月はユーザーインタフェイスに焦点を合わせてみよう。

2ボタンスティックの操作性

餓狼SPは，オリジナルのNEO・GEO版では8方向ジョイスティック&4ボタンスティックで操作する。X68000版はキーボード，2ボタンスティック，前作「餓狼伝説2」(以下，餓狼2)に付属していた4ボタンパッド，そしてセガのメガドライブ用6ボタンジョイスティック(ジョイパッド)に対応している(詳細は後述)。

さて，多くのX68000ユーザーは2ボタンジョイスティックを所有しているようだが，このスティックで，どのようにして4ボタンの操作性を実現するのか。NEO・GEO版では4ボタンはそれぞれパンチ/キックの強攻撃，弱攻撃の4種類のアクションに割りふられている。X68000版の2ボタンでの操作は，ボタン2つを物理的にパンチとキックに割り当て，長く押すと強，短く押すと弱，となっている。また，超必殺技には「弱パンチボタンと強キックボタンの同時押し」のような複雑な操作系を必要とする場合があるが，これは2つのボタンを同時押しというように操作は簡略化された。

X68000用　5″2HD版　9,800円(税別)
魔法株式会社　☎078(261)2790

さて，実際にこのX68000版餓狼SPを2ボタンのジョイスティックでプレイしてみると，残念ながら操作性はかなり悪い。前作の餓狼2のほうが「マシ」だったと思わずぼやいてしまうほど悪い。残念ながらこれでは餓狼SPはまともに遊べない，と私は判断する。

まず，「レバーをしばらく引いてから……」という操作が必要な「溜め系必殺技」がことごとく出ない。ボタン2つ同時押し操作の超必殺技もまったく出ない。かなりやり込んだのだが，超必殺技は結局一度も出なかった。その後，Oh!Xスタッフ数名でいろいろやってみたところ，溜め系の技は，溜めたのちにレバーを溜め方向に入れたまま一足先にボタンを押し，遅れて溜めを開放するレバー操作をするとうまくいくことが判明した。たとえば，キムの飛燕斬ならばレバーをしばらく下に引き，溜まったらキックボタンを押して，やや遅れてレバーを上へ，といった具合だ。しかし，これではオリジナルとは全然違うプレイ感覚だ。

6ボタンスティックを用意せよ

一方，6ボタンスティックを使ってプレイしたところ，オリジナルとまったく同じ操作感覚が実現された。溜め技も楽々出るし，通常技からのキャンセル超必殺技もザクザク。鳳凰脚を出して滑り寄るキムをテリーがパワーゲイザーで撃退，なんていうダイナミックシーンが見られるようになった。2ボタンでのプレイとはまるで違う爽快レスポンス！　そう，まるで別のゲームのようだ。ここで「魔法株式会社製X68000版『餓狼SP』は6ボタンスティックでプレイするようにプログラムされている」と言い切ってしまうことにする(もちろん，餓狼2に付属のパッドでもいい)。

で，6ボタンジョイスティックだが，これは先述のとおりメガドライブ用の6ボタンスティックをX68000に繋ごうというものだ。接続には電波新聞社のアクションゲーム「チェルノブ」に付属しているアダプタが必要ということになっているが，それがなくても実は簡単に繋ぐことができる。ここで，その6ボタンスティックの接続方法を紹介しよう。

偶然にもメガドライブ用ジョイスティックのコネクタはX68000のものと同じ形状をしている。繋いでみると8方向&1ボタンスティックとして使用できるが，そのままでは餓狼SPはプレイできない。

そこで，変換コネクタ(写真1)を自作することとする。図1がその製作方法である。部品はD-SUB9ピンのオスとメスのコネクタとコードだけ。部品代は200～300円程度で済むだろう。工作なんかしたことがない人でも大丈夫だ。作り方はいたって簡単で，図のようにオスとメスのコネクタをコードで配線するだけ。ピンとピンの間隔が狭い

図1　変換コネクタの製作方法

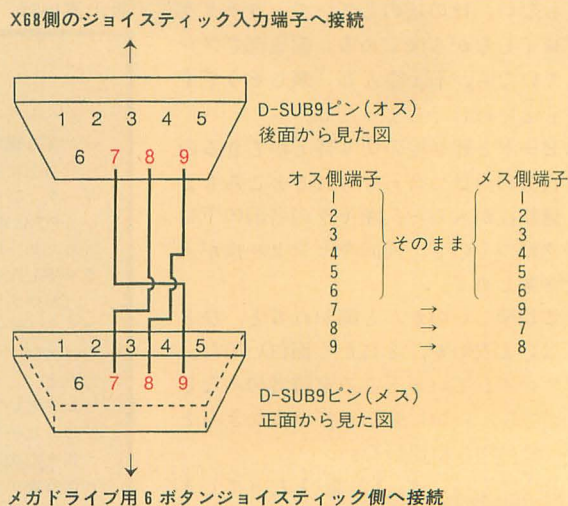
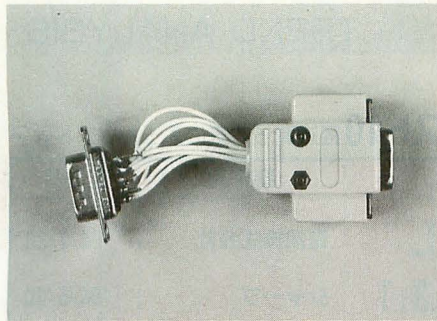



写真1　変換コネクタ

ので隣のピンへ電線が接触しないようにする点だけに注意すればいい。これで，6ボタンスティックで快適な操作感覚を味わえる。

西川善司お勧めのジョイスティックはHORI製の「FIGHTING STICK MULTI」で，スティック，ボタンの感触・反応とも最高だ。カプコン製の「CPSファイター」はボタンとボタンの間隔が狭くて大人の手と相性の悪い「お子様向け」だと思われるのだが，この，HORI製のは若造からオジサンの手にもちゃんとフィットしてくれる。お試しあれ。

ジョイスティックによってはボタン配置が餓狼SPに適したものとなっていない場合がある。その場合は餓狼SPの「オプション」のジョイスティックセットアップメニューで希望のボタン配置を設定してほしい。

サウンドあれこれ

今回の餓狼SPでは，前作X68000版「餓狼2」よりもサウンドの完成度も上がった。餓狼2ではカットされていたPCMフレーズも見事に再現され，オリジナルに肉薄した雰囲気を楽しめる。MIDIはSC-55とMT-32の両方に対応で，FM音源，AD PCM音源とのアンサンブルにより原曲に忠実なゴージャスサウンドを実現している。Z-MUSIC対応なので曲データファイルだけあとでこっそり楽しむなんてことも可能。

効果音は，X68000が単音AD PCM音源のハードしか持ち合わせていないせいもあり，バイクや汽車の走行音，川のせせらぎなどはカットされている。前作では，BGMをSC-55で演奏させた場合はSC-55にある効果音を駆使してそういった音響演出も一部再現されていたのだが。舞のステージの川のせせらぎが，今回は省略されているのは謎。ちょっとさびしい。

図2　PCM8.Xを先に組み込む起動バッチファイル

```
pcm8
zmsc -m -U -t0 -p80 -o4 -b GDSP_MH.ZPD
gspmain_.x %1 %2
```

音響制御にZ-MUSICを使用していることもあって，PCM8.Xなどの多重PCM音源ドライバを先に組み込めば，PCMドラム/パーカッション，PCM効果音が同時再生され，アーケード気分が楽しめる。アンディが「昇龍弾」と叫び，ジョーの「タイヤキィ」がこれに被り，BGMではエキゾチックな男性コーラスタイが怪しげな歌声を張り上げる……なんてことが実現しちゃうのだ。参考までに，起動バッチファイルの例を示しておこう(図2)。

しかし，現実には，これを実現するにはX68030クラスで画面モードを31kHzにしていないと難しい。それ以下の環境では，一応，多重サウンドにはなるものの，音がブチブチ切れてしまう。原因は餓狼SPのプログラムで，解析した感じでは，VDISP割り込み処理とCRTCRAS割り込みの処理のルーチンに問題があるようだ。技術的な話をすると，多重PCMドライバを使用する場合，多重PCMドライバ以外のプログラムは多重PCMドライバの割り込みを最優先させなければならないのだが，餓狼SPのプログラムではこのようにはなっていない。したがって，VDISP割り込み処理中に多重PCMドライバの割り込みが発生した場合，多重PCMドライバの処理が待たされ，結果的に音が切れる……ということだ。X68030級だと，それぞれの処理時間がわずかなので割り込みがかち合う状況は発生しにくい。また31kHzだと，VDISP割り込みの発生間隔が15kHzよりも長いので，同様に割り込みがかち合いにくいのである。とはいえ，X68030級で31kHz画面モードでも完璧ではない。

多重PCMドライバを使うのはユーザーの勝手でメーカーの意図するものではないのだから文句はいえないが，そういう配慮もほしかったものである。ちなみに，パソコン通信ネットワークPC-VANのX1CLUB(ジャンプコードJX1)に私が作成した餓狼SPのVIDSP割り込み，CRTCRAS割り込みの各処理ルーチンを多重PCMドライバに適応させるパッチプログラムがアップされているので，興味のある方は覗いてみてほしい。

紫パンツはホモのしるしとの説も(?)

グラフィックあれこれ

キャラクターのアクション，背景動画ともにほぼ完璧に再現されている。

グラフィック機能という観点では，X68000はNEO・GEOに比べて圧倒的に劣っている。そのため，背景はオリジナルからはかなりの減色処理を施しているそうだが，パッと見た感じではそんな「色落ち」を感じさせない素晴らしいものになっている。前作で不評だった奥ラインで崩れるキャラクターグラフィック現象&インチキ拡大縮小も改善された。作品を発表するたびに「魔法」は確実に技術が進歩しているようだ。

欲をいえば，グラフィックはお決まりの256×256モードではなくて，カプコンのX68000版「ストリートファイターIIダッシュ」や電波新聞社のビデオゲームアンソロジーシリーズのような384×256モードのアーケードライクな解像度で移植してほしかった。これは次回以降の移植作品への期待ということで，よろしく魔法さん。

おわりに

とにかく「餓狼伝説SPECIAL」は6ボタンスティックを接続して初めてその真価が発揮される。ということで，ぜひとも，環境を整えて，対戦にCPU戦に，各自熱いバトルを展開してくれ。

そのほかの耳より(?)情報

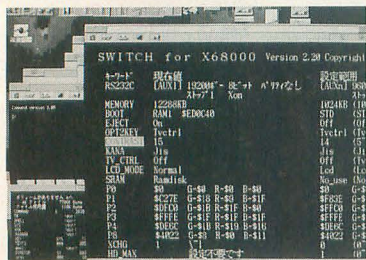
ゲームプログラムは，容量を200Kバイト以上も増やすことのできる9SCDRVという特殊なFDDドライバを使ってフォーマットしたフロッピーディスクに収められている。しかし，どうもこのフォーマットと相性の悪いドライブもあるようで，ハードディスクへのインストールや起動がときどき失敗する，という報告が届いている。そういう事態に陥ったときは，焦らず「(R) 再実行」を押そう。しばらく「再実行」しているとうまくいくはず。

あと，本文で述べた以外にもうひとつ要望があった。セットアップで設定したボタン割り当てや音源，解像度などの各種設定は，記録保持してほしかった。せめて，ハードディスクにインストールしている場合だけでも……。ゲーム始める前にいちいち設定するの面倒なのねーん。

ちなみに，キーディスクをドライブに挿入しておけばハードディスクから起動できる。こりゃいいね。そういえば，SHIFTキー押しながら起動するといいことあるかもねーん。

1994年9月号のハガキ集計ベスト10 最近買って気に入ったソフトは？

POINT	タイトル	発売元	発売日
112	餓狼伝説SPECIAL	魔法株式会社	'94/7/28
73	SX-WINDOW ver.3.1	シャープ	'94/5/30
46	ストリートファイターIIダッシュ	カプコン	'93/11/26
23	大魔界村	カプコン	'94/4/22
15	クイーン・オブ・デュエリスト外伝α＋	TAKERU	'94/8/10
15	ジオグラフシール	エグザクト	'94/3/15
12	レッスルエンジェルス3	TAKERU	'94/7/31
12	あすか120% BURNING Fest.	ファミリーソフト	'94/4/22
12	宝魔ハンターライム	TAKERU	'94/7/10
10	悪魔城ドラキュラ	コナミ	'93/7/23
10	Mr.Do！/Mr.Do！ vs UNICORNS	電波新聞社	'94/7/2

（無作為抽出した1000通のハガキを集計）

2カ月ぶりの集計なので，当然ながら状況は大きく変わっています。

まず，7月の末に発売された「餓狼伝説SPECIAL」が予想どおり1位。期待に違わぬ出来にファンも満足したという結果がこの集計にも表れています。前作からの技術進歩への評価から，メーカーへの感謝と共に早くも次回作への期待の眼差しも注がれています。Oh!Xでも先月号に引き続きレビュー記事を掲載していますので，まだプレイしていない人もぜひ読んでくださいね。

3位の「ストリートファイターIIダッシュ」は前回10位(10ポイント)から大幅に増えて大躍進(？)となりました。これは，次作「スーパーストリートファイターII」の発売決定の影響が大きいようで，「期待のソフト」「満足したソフト」にこの2作を挙げている人が多く見られました。スーパーに備えてダッシュで腕を磨いているのでしょうか。遊んでいるうちに素晴らしさを再確認した，という感じのようです。ついでというわけではないでしょうが，同じくカプコンの「大魔界村」も順位アップしました。もっとも，こちらはポイント数は横這いで固定ファンによるポイント獲得のようですが，それだけ飽きずにプレイされているということでしょう。

そのほかのゲームについては，ポイント数もほとんど変わらす横並び状態になっています。上位のソフトにポイントが集中したのも一因でしょうか。ゲームの完成度などによる評価というよりは，単純に好みにより分かれているというのが正しいようで，獲得ポイント数があまり高くない割には，各人の思い入れ度や熱中度の高さが見受けられるはがきが目立ちました。

1年間続いた「宝魔ハンターライム」もランキング入り。全12話がとうとう完結してしまい，ファンには寂しい限りですが，最終話の5枚組という豪華さには満足度も高かったようですね。キャラクターの個性とアニメーションに加えて，お手軽さと低価格も好評でした。

それにしても根強い人気の「悪魔城ドラキュラ」。1年を超えるランキング入りで記録更新中ですが，今後，これを超えるソフトは登場するでしょうか。各メーカーに期待したいですね。

[特別企画]

もみじ狩りPRO-68K

どうやら今回も無事に付録ディスクをお届けすることができたようだ。今回の付録ディスク「もみじ狩りPRO-68K」は展開後，ディスク3枚分になる。別に詰め込み方が甘くなってきているわけではない。LHAのプログラム圧縮率を考えると，本来はこんなものになるはずなのだ。いつも痛感することながら，1.2Mバイトは狭い。
SLASHはすでに完全なかたちで収録することが困難になっているのはご承知のとおりで，さらにSX-BASICのシステムもかなり巨大なものになってきている。今回はSLASHのモデラはソースファイルを省略して実行ファイルのみ，SX-BASICはソースと実行ファイルを両方とも収録，というかたちにした。理屈でいえばSLASHを使うということは，すなわちSLASHの開発環境を持っていなければならないのだが，2Mバイトではコンパイルできないこととか，開発環境がなくても動作くらいは見てみたいというのももっともなことから実行ファイルのみの収録となった。
1枚のディスクの可能性を追究するスタイルから始まったOh!Xの付録ディスク。そろそろ転機を迎えなければ苦しいのだが，まだ新しいかたちが見つからない。試行錯誤を兼ねて，もっと毛色の変わったディスクを作ってみたいものだ（とは思う）。

CONTENTS

PRO-68K

収録プログラム&データ解説

付録ディスクの使い方

編集部

もうすっかりお馴染みの付録ディスクをお届けします
今回は圧縮のようすがいつもと違うので注意してください
それでは，さっそく中身を紹介していくことにしましょう

ひなまつり，こいのぼり，GENIEに続き，今年4つめの付録ディスク「もみじ狩りPRO-68K」をお届けします。

展開後のディスク枚数はやや控えめですが，それでも圧縮ファイル自体の容量は1.4Mバイトフォーマットを使った秋祭りPRO-68Kに次ぐものとなっています。フロッピーの狭さを再認識する今日この頃です。

当初は地味に「小物集」という構想があったのですが，結局のところ，やはり大物が大方を占めてしまいました。それも大半がSX-WINDOW関係のアプリケーションとなっています。SX-WINDOW関係のツールはファイルサイズが大きくなりがちということもありますが，まあこれも時代の流れというものでしょう。

ディスクの展開は1Mバイトでも大丈夫です。念のために「要2Mバイト」としておきました。しかし，SX-WINDOW関係がメインですから実質的に「要4Mバイト」となりつつあります。

ディスクの展開について

付録ディスクはバッチファイルによりフロッピーディスク3枚に展開されます。あらかじめフォーマットしたフロッピーディスク3枚を用意しておいてください。あとは付録ディスクを0ドライブに入れてX68000を起動し，表示されるメッセージにしたがって操作すれば自動的にファイルが展開されます。

ディスクに収録されたファイルの圧縮にはお馴染みのLHA.Xを使用していますが，今回はディスク容量の都合によりLHA.Xが収録されておりません。

各自で必要なファイルだけを展開したいという場合もあるでしょうが，できるだけフロッピーディスクにすべて展開しておくことをおすすめします。

どうしてもという場合，DISKn.Xは，LHA.XでLZHファイルと同様に扱えますので，以下のような手順で展開してください。

例）　MODを展開したいという場合

A>LHA E MOD.X MOD*.*

また，DISKn.Xは直接実行するとカレントディレクトリに，圧縮されているファイル群を展開します。

たとえばDISK1.Xを実行するとカレントディレクトリに，

XBASFNC
CALCOMP
PUSH
MOD
HCOPY
MANDEL

という6つのディレクトリと，

回路.LZH

を作成します。

付録ディスクに収録されている内容は表1のとおりです。参考にしてください。

ディスク1の内容

ディスク1はSX-WINDOW以外で使用するツールを中心にまとめてあります。

●X-BASIC用外部関数

XSPRITE.FNC（伊藤雅彦）

スプライトゲームを支援する高位関数です。キャラクター管理，仮想画面マップスクロール，ヒットチェック，スティック入力などを行います。

コンパイルにも対応していますので，X-BASICでもそこそこ本格的なゲーム作成ができるようになります。

EXEC.FNC（江川乃誉司）

要するにX-BASICで8ビット時代のようにマシン語を扱うための関数です。作者は「これでショートプロが派手になる」と主張しております。

XSPRITEでも足りない部分などを補足する最終兵器といえるでしょう。サンプルゲームつきですので遊んでみてください。

TGRAPH.FNC（朝倉祐司）

テキスト画面への描画関数です。サポートされている関数は，

TCLS,TLINE,TFILL,TPSET,
TPUT,THOME, TPALET,TPOINT

の8種類です。使い方はサンプルプログラムをご覧ください。

●PUSH BON！追加マップ集

こいのぼりPRO-68Kで発表したパズルゲームPUSH BON！のユーザーデータ募集での応募作品をまとめたものです。マップデータだけですので，使用するのはPUSH BON！本体が必要になります。

おまけとして，いろいろお騒がせした前作36面の解法（39手）つきです。

●タブレットマウスドライバ（菊地功）

NSカルコンプ社製のドローイングパッド，ドローイングスレートをマウスの代わりに使用するためのデバイスドライバです。

●カラーハードコピーツール（瀧康史）

シャープIO-735シリーズ，エプソンMJ700V2Cに対応した印刷プログラムです。

かなり実行時間がかかりますので注意してください。

●SX-MANDELBROT（中森章，田村健人）

事情により未完成版ですが，SX-WINDOW上でマンデルブロ集合を表示するプログラムです。指定した範囲を拡大表示できます。あまり拡大率を上げると非常に精度が悪くなることがあります。

このプログラムを起動するときは壁紙動画.Rの常駐解除を行ってからにしてください。表示座標がずれます。また，アイコン化はしないでください。

●回路.DLB（瀧康史）

EasydrawSX-68Kで回路図を書くためのライブラリ集です。

LHA E　回路.LZH

として解凍してからお使いください。

ディスク2の内容

SX-WINDOW用各種ツールをまとめたものです。

●SX基本ツール（田村健人）

2行にするの.R

ディレクトリウィンドウのファイル名表示で長いものを2行に分かち書きします。

アイコン間隔.R

ディレクトリウィンドウのアイコン表示間隔を任意の幅に設定します。

ARLK.X

高性能なリソースリンカです。

ESYSD.X

9月号で発表したデスクトップの起動情報エディタです。

●SADJUST.R（福嶋章太）

多彩な画面解像度に対応するためのツールです。各種拡張機能を含みます。

●壁紙動画.R（福嶋章太）

9月号で発表したSX-WINDOWの背景でアニメーションを実行するツールです。

●EXCOM（田村健人）

シャーペン用の外部コマンド群です。C言語支援用コマンドや，文書中のアイコンをボタンとして機能させるコマンドなど11種を収録しています。

●ベル.X（石田伯仁）

ひなまつりPRO-68Kで発表したイベントに応じてAD PCM音を再生するツールのバージョンアップ版です。

●SX-BASIC用ゲームキット（田村健人）

SX-BASICから使用できるゲーム専用エンジンです。

フィールドタイプのRPGなどを作成するためのフィールド.X,3D迷路を管理するMAZE3.X,各種情報を表示しユーザーインタフェイスを提供する文字画面.Xの3種がサポートされました。

ディスク3の内容

●SX-BASIC暫定版その3（石上達也）

お馴染みSX-BASICのバージョンアップ版です。今回はマルチフォントテキスト対応やグラフィックスクリプト対応など表現力の強化のほか，新たに，コードリソース呼び出しによる拡張手順やリソース切り換えによるデータ共用などにも対応しています。従来のプログラムを動作させるには若干プログラムの変更が必要ですので注意してください。

●SX-BASIC用ゲーム各種（郡茂樹）

読者投稿によるなかなか気合の入ったゲームプログラムです。

関西は要するに上海の流れを汲むゲームです。特にルールは説明するまでもないでしょう。ちなみにファイルの入出力でコケるとしたらSX-BASICのせいです。まだ対応できていません。ごめんなさい。

そのほか，ついにSX-BASICでカードゲームができるようになりました。収録されている3つのカードゲームは郡さんの作品を編集部でリソース切り換えに対応させたものです。SX-BASICプログラマの方はぜひ参考にしてみてください。

*　　　*　　　*

ひょっとすると年内にもう1回……という，定期購読の人は笑いが止まらない展開になるかもしれませんが，あくまで予定は未定ですのであまり期待せずにお待ちください。

そのほか，今回発表したツールを使った投稿なども広く募集していますのでそちらもよろしくお願いいたします。

表1　収録プログラム一覧

```
Listing of Archive : e:DISK1.X
<< dir >> xbasfnc/exec/
 SHOCK              .BAS
 ENEMY              .DAT
 EXEC               .FNC
 EXP                .PCM
 HIT                .PCM
 IGN                .PCM
 SHOT               .PCM
 ARGSET             .R
 DLC                .R
 ARGSET             .S
 DLC                .S
 EXEC               .S
 JOYGET             .S
 SOUND              .S
 SP_SETS            .S

<< dir >> xbasfnc/xsprite/source/
 xsp_def            .h
 xsprite            .s
 xsp_work           .s
 xsp_int            .s
 sp_xinit           .s
 sp_slon            .s
 sp_sloff           .s
 sp_slidp           .s
 sp_slidv           .s
 sp_loc             .s
 sp_hang            .s
 sp_leave           .s
 sp_hgadd           .s
 sp_hgrmv           .s
 sp_hitrng          .s
 sp_hiton           .s
 sp_hitoff          .s
 sp_hit             .s
 sp_stkon           .s
 sp_stkoff          .s
 sp_xstat           .s
 sp_intcnt          .s
 bg_slide           .s
 bg_map             .s
 bg_mpscrl          .s
 bg_xstat           .s
 xstrig             .s
 scrpri             .s

<< dir >> xbasfnc/xsprite/BtoC_kit/
 XSPRITE            .DEF
 XSPRITE            .H
 xsp_work           .o
 xsp_int            .o
 sp_xinit           .o
 sp_slon            .o
 sp_sloff           .o
 sp_slidp           .o
 sp_slidv           .o
 sp_loc             .o
 sp_hang            .o
 sp_leave           .o
 sp_hgadd           .o
 sp_hgrmv           .o
 sp_hitrng          .o
 sp_hiton           .o
 sp_hitoff          .o
 sp_hit             .o
 sp_stkon           .o
 sp_stkoff          .o
 sp_xstat           .o
 sp_intcnt          .o
 bg_slide           .o
 bg_map             .o
 bg_mpscrl          .o
 bg_xstat           .o
 xstrig             .o
 scrpri             .o
 lib_ind            .txt

<< dir >> xbasfnc/xsprite/
 PEEKPOKE           .FNC
 XSPRITE            .FNC
 xsp                .bas
 xsp_10MHz          .bas
 maptest            .bas
 xsprite            .doc
 xsp                .x

<< dir >> xbasfnc/tgr/
 sample             .bas
 sample             .x
 tcls               .s
 tfill              .s
 thome              .s
 tline              .s
 tpalet             .s
 tpoint             .s
 tpset              .s
 tput               .s
 tgraph             .a
 tgraph             .def
 tgraph             .h
 tgraph             .fnc
 tgraph             .bat

<< dir >> calcomp/
 HIOCS              .EQU
 CalTab             .s
 CalTab             .mak
 KEEPCHK            .S
 PSPDEF             .H
 CalTab             .x

<< dir >> push/
 PUSH_1             .MAP
 高橋               .DOC
 PUSH_2             .MAP
 三沢               .DOC
 PUSH_3             .MAP
 PUSH_4             .MAP
 関東               .DOC
 PUSH_5             .MAP
 加藤               .DOC
 PUSH_6             .MAP
 益山               .DOC
 PUSH_7             .MAP
 STAGE36            .DOC
 READ               .DOC

<< dir >> mod/
 mod                .x
 plgconv            .x
 slash              .PLG
 BOSS_SEN           .PLG
 SHIP_ALL           .PLG
 ENEM12_2           .PLG
 ENEM8              .PLG

<< dir >> hcopy/
 hcescv2x           .x
 hcescv2x           .c
 makefile
 hc735              .c
 hc735light         .c
 readme             .doc
 hc735light         .x

<< dir >> mandel/
 cfilter            .c
 menu               .c
 sub                .has
 v3                 .has
 m                  .c
 Makefile
 m2                 .c
 まんでる風呂       .X
 readme             .doc

<< dir >> / (root)
 回路               .Lzh

Listing of Archive : e:DISK2.X
<< dir >> excom/
 Makefile
 e_c_brace          .c
 e_c_brace          .ex
 e_c_sharp_sign     .c
 e_c_sharp_sign     .ex
 excom              .h
 freturn            .c
 freturn            .ex
 i_f_comment        .c
 i_f_comment        .ex
 isearch            .c
 isearch            .ex
 optab              .c
 optab              .ex
 rpar               .c
 rpar               .ex
 setkind1           .c
 setkind1           .ex
 tohan              .c
 tohan              .ex
 wopm               .c
 wopm               .ex
 xclick             .c
 xclick             .ex
 xpaste             .c
 xpaste             .ex

<< dir >> sx/
 esysd              .x
 arlk               .x
 2行にするの        .pen
 2行にするの        .r
 PAT40046           .pt4
 PAT40047           .pt4
 アイコン間隔       .pen
 アイコン間隔       .r
 arlk               .doc

<< dir >> sadjust/
 SAdjust            .r
 SAD_GET            .sxb
 SAdjust            .s
 SAdjust            .pen

<< dir >> 壁紙動画/pat_make/
 a                  .bat
 b                  .bat
 c                  .bat
 d                  .x
 readme             .doc

<< dir >> 壁紙動画/
 壁紙動画           .s
 sxpics             .x
 picslice           .h
 picslice           .c
 sx                 .c
 sxpics             .h
 x                  .PIC
 gracon             .sxb
 壁動玉々           .s
 壁紙動画           .r
 壁紙動画           .pen
 壁動玉々           .r

<< dir >> bell/他のソースリスト/
 settei             .has
 timebtn            .s
 testplay           .s
 FreqBtn            .s
 Button             .s
 WDEFicon           .has
 kankyo             .has
 AboutMe            .has
 aboutpat           .s

<< dir >> bell/
 ベル               .x
 ベル               .LB
 ベル               .Pen
 ベル               .has
 bell               .h
 JoyA               .s

<< dir >> sxgamekit/
 フィールド         .lb
 フィールド         .x
 文字画面           .x
 Makefile
 anames             .c
 anamesm            .c
 d9                 .h
 draw               .c
 drawm              .c
 f                  .c
 fockfield          .sxb
 fockmaze3          .sxb
 fockmg             .sxb
 initm              .c
 j10k               .app
 joy10key           .r
 mapm               .c
 maze3              .h
 maze3              .lb
 maze3              .x
 mg                 .c
 mknullmap          .c
 mknullmap          .x
 mnutil             .c
 rgnm               .c
 send               .c
 sendm              .c
 sendmes            .c
 sendmesm           .c
 skelm              .c
 wmfore             .app

Listing of Archive : e:DISK3.X
<< dir >> sxb/sxbasic/
 machine            .s
 comp               .c
 dialog             .c
 exp                .c
 file               .c
 inter              .c
 sx                 .c
 sbedit             .c
 efunc              .c
 sbfile             .c
 sxbasic            .h
 table              .c
 makefile
 新幹線             .sxb
 SxBasic            .x

<< dir >> sxb/edit/
 MAKEFILE
 TESCROLL           .C
 tedit              .c
 TEDIT              .H
 temain             .c
 machine            .s
 bedit              .x

<< dir >> sxb/wind/
 makefile
 sdbmp              .h
 windf              .c
 windc              .c
 windb              .c
 winde              .c
 wind               .c
 wind               .h
 wind               .lb
 windd              .c
 machine            .s
 wind               .x

<< dir >> sxb/engine/
 engnd              .c
 engnc              .c
 engn               .c
 engnf              .c
 engn               .h
 makefile
 engine             .x

<< dir >> sxbas_game/
 CARD               .LB
 関西               .LB
 関西               .DAT
 MakeData           .SXB
 関西               .SXB
 神経衰弱           .LB
 神経衰弱           .PT4
 花占い             .LB
 花占い             .PT4
 ﾋﾟﾗﾐｯﾄﾞ            .LB
 ﾋﾟﾗﾐｯﾄﾞ            .PT4
 神経衰弱           .SXB
 花占い             .SXB
 ﾋﾟﾗﾐｯﾄﾞ            .SXB
```

SX-BASIC公開デバッグ第7回

SX-BASIC(暫定版その3)

Ishigami Tatsuya 石上 達也

だんだんと整備されてくるSX-BASICプログラミング環境
より豊かな表現力と周辺ツールとの連係プレイ
さらにコードリソースによる拡張などがサポートされています

付録ディスクにはSX-BASICの最新版が収録されています。これでSX-BASICで予定されていた機能の8割くらいはサポートされたことになると思います。今後の展開を考えるとまだ流動的な部分や試験的なものも含まれています。シャーペンとのリンクやページ機能の追加などの課題も残されていますが，今月号の付録ディスクに収録されたプログラムを見ると，どうやらコンパイラの作成を急ぐほうがよいのではないかという状況にもなってきました。

今回はソースリストつきですので，皆さんにはさらに突っ込んだデバッグをお願いしたいと思います。また，コンパイラ作りなら任せとけという方がいらっしゃいましたら，ぜひ編集部までご一報ください。

また，今回のディスクではSX-BASICに対応したプログラムも多く収録されていますので，関連プログラムについてもよく解説を参照するようにしてください。

ラインエディタがなくなった

それでは今回追加/変更された部分を解説していきましょう。

「こいのぼりPRO-68K」までのバージョンではラインエディタを用いてプログラミングを行っていました。直接，命令を入力するとダイレクトコマンド，行番号の次に命令を入力するとプログラム，というふうに振り分けられていたのです。

今回のバージョンでは，試験的にマルチフォントテキストに対応した関係上，ラインエディタを廃止しました。現状では以下のように，SX-BASICにプログラムを読み込ませてください。

1) SX-BASICの起動時に，ファイル名をパラメータとして与える

　－Fファイル名

で起動時に読み込みファイルを指定できます。SX-BASICが起動すると，ただちにプログラムの実行が開始されます。

例) SXBASIC -Fmelodius.sxb

2) シェル上からファイルをドラッグしてくる

すでに起動しているSX-BASICのウィンドウにファイルアイコンを放り込むことによってファイルを指定することができます。

いずれかの方法により，SX-BASICへプログラムを読み込ませることができます。

以前はSX-BASICにプログラムを読み込ませることは「list」命令で中身を見られるようにすることでしたが，バージョン0.5からはラインエディタが廃止された関係上「list」命令がありません。当然，「run」もありません。ファイルが指定されたら，即，読み込み/(中間コードへの)コンパイル/実行開始という一連の作業が行われます。

SX-BASICのウィンドウには，エディタ環境としての機能はまったくなくなったのです。

エディタ機能がなくなって，残っているのは，イミディエイトウィンドウとしての機能だけです。イミディエイトウィンドウというのは，print文やinput文などの入出力先として用いられるウィンドウのことです。SX-BASICは，本来ユーザーインタフェイス部分をそっくりウィンドウエンジンに任せているのですが，ちょっと変数の中身を確認したいときなどに例外として用いられるウィンドウです。

ウィンドウエンジンで文字列をやりとりするには，テキストアイテムや各種コントロールのアイテム（ボリュームやアップダウンボタンなど）を用いますが，使用に先だって，それらのアイテムを準備しておかなければなりません。

デバッグ中など，ちょっと変数の値を見たいときなどに，いちいちアイテムを用意するのは不便ですし，アイテムを用意すると実行環境が変化してしまい，バグの出方が変わってしまう場合もあるかもしれません。

そんなときには，print文やinput文で簡単に入出力が行えるSX-BASICのイミディエイトウィンドウを用います。

また，デバッグ時だけでなく，ウィンドウエンジンを用いるまでもない小規模なプログラムでは，すべての入出力をSX-BASICから行うこともできます。

グラフィック&スクリプトに対応

今回のバージョンからは，ウィンドウエンジン内に，線分や円などのグラフィックを描画できるようになりました。また，これらの描画過程はSX-WINDOWのPICT形式でスクリプトとして保存されますので，カット&ペーストを行い，図形をEasydrawなどへ持っていくこともできるようになりました。

グラフィック描画はウィンドウアイテムかビットマップアイテムへメソッドを送りつけることによって行われます。現在のところ，用意されているメソッドには以下のような種類があります。

●pensize

ペンサイズを設定します。上位ワードが水平方向のペンの太さ，下位ワードが垂直方向のペンの太さを表します。

```
Bitmap1.pensize = &h00010001
```

ペンサイズを（1，1）の大きさに設定します。

●penmode

ペンモードの設定を行います。設定内容は以下のとおりになっています。

上位バイト

　0　フォアグラウンドカラーで描画
　1　バックグラウンドカラーで描画

下位バイト

　0　pset
　1　and

2　or
3　xor
4　npset
5　nand
6　nor
7　nxor

例)　Bitmap1.penmode ＝ 0

注）SX-WINDOWでは，上位バイト＝2でペンパターンで描画，3でエクステンドパターンで描画となっていますが，現在のウィンドウエンジンの仕様では，これらのパターンがどのように設定されているかは不定です。よって，上位バイトに2と3を指定しないでください。

●**forecolor**

描画色を設定します。設定内容は以下のとおりになっています。

0：透明
8：白
9：薄いグレー
10：濃いグレー
11：黒
12：黄
13：赤
14：緑
15：青

注）テキストアイテムなどでは，SX-BASIC独自のカラーコードを用いています。しかし，ビットマップアイテムの指定にはSX-WINDOWシステム規定の値を用いています。両者の間に互換性はないので注意してください。

例)　Bitmap1.forecolor ＝ 11

●**backcolor**

背景色を指定します。設定内容はforecolorと同一です。

例)　Bitmap1.backcolor ＝ 9

●**box=x1,y1,x2,y2**

アイテム内の(x1,y1,x2,y2)で表されるレクタングルを，ペンモードに従って描画します。座標はアイテム内のローカル座標で指定します。アイテムの大きさに合わせてクリッピングが行われます。

例)Bitmap1.box ＝ 10,10,20,20

●**circle=x,y,r**

アイテム内に（x,y)を中心に半径rの円を描画します。座標はアイテム内のローカル座標で指定します。アイテムの大きさにあわせてクリッピングが行われます。

例)　Bitmap1.circle ＝ 10,10,5

●**fill=x1,y1,x2,y2**

アイテム内の(x1,y1,x2,y2)で表されるレクタングル内を，ペンモードに従って塗りつぶします。座標はアイテム内のローカル座標で指定します。アイテムの大きさに合わせてクリッピングが行われます。

例)　Bitmap1.fill ＝ 10,10,20,20

●**line**

アイテム内に(x1,y1)，(x2,y2)を端点とする線分を描画します。座標はアイテム内のローカル座標で指定します。アイテムの大きさに合わせてクリッピングが行われます。

例)　Bitmap1.line ＝ 10,10,20,20

スクリプトの扱い

今回拡張されたメソッドを用いて描画を行うと，その結果はPICTと呼ばれる形式のスクリプト（script：手順書，台本）に収められます。このスクリプトを収めたハンドルは「script」プロパティにて参照できますので，

```
clipboard.pict ＝ Bitmap1.script
```

とすることで，クリップボードへ図形をコピーすることができます。

いろいろと描画を行っていくと，以前の図形が邪魔になってくるかもしれません。

```
for i＝ 0 to 4
  Bitmap1.forecolor ＝ 9
  Bitmap1.line=0,(i-1)＊4,40,
                        (i-1)＊4
  Bitmap1.forecolor ＝ 11
  Bitmap1.line ＝ 0,i＊4,40,i＊4
next
clipboard.pict ＝ Bitmap1.script
```

このプログラムを実行すると，横線がすうっと下へ落ちていく様子が表示されます。これは，一度描いた線分を薄い灰色で消し，その直後に，同じ大きさの線分を下にずらして表示することによって，アニメーション効果を出しています。

アニメーションの動作がひととおり終わった時点で，クリップボードへ転送を行うと，このアニメーションごと送られてしまいます。動くデータを送りたい場合はこれでよいかもしれませんが，当然のことながら，最終的に得られたスクリプトのみを送りたい場合もあります。

一度描いて消された線分がスクリプト内に残っているから，このような事態が起こってしまうのです。適当なタイミングで過去の記録を消去してやれば，このような問題は起きないでしょう。

スクリプトの記録を消去するには，scriptプロパティに0を代入してやります。

例)　Bitmap1.script ＝ 0

先ほどの例では，

```
for i＝ 0 to 4
  Bitmap1.forecolor ＝ 9
  Biymap1.line=0,(i-1)＊4,
                        40,(i-1)＊4
  Bitmap1.script ＝ 0
  Bitmap1.forecolor ＝ 11
  Bitmap1.line ＝ 0,i＊4,40,i＊4
next
clipboard.pict ＝ Bitmap1.script
```

とすることにより，アニメーションの最後の状態のみをクリップボードに転送することができます。

resizeプロパティ

スクリプトに収められているのは，基本的にドローデータです。ある領域内を点（ドット）の集まりとみなすのではなく，

点(X1,Y1)から点(X2,Y2)まで線を引く

というようなデータ形式で，図形を扱います。ディスプレイに表示するにしろ，プリンタに打ち出すにしろ，最終的には点の集まりとして出力するのですが，その直前段階までドローデータ形式で保持します。ですから，拡大/縮小を行っても画像の劣化が起きにくいという特徴を持っています。

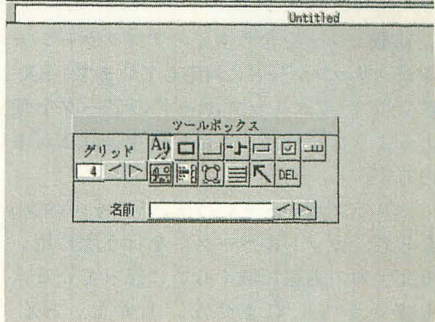
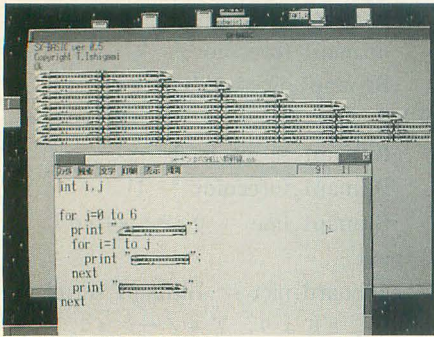
この拡大/縮小を制御するのがresizeプロパティです。resize＝0のとき，図形は描画時とまったく同じ大きさを保持します。0以外のときは，描画時のアイテムの大きさに合わせて拡大（現在のアイテムが大きくなったとき）/縮小（同小さくなったとき）を行います（図1を参照)。

「クリップボード」でPICTデータを表示するとメニューで「そのままの大きさで表示」(?)，「ウィンドウサイズで表示」という項目が表示されますが，前者がresize＝0，後者がresize≠0の場合に相当します。

このプロパティは，PICTデータに与えられるのではなく，アイテムに対して与えられています。ですから，コピーされた

図1：ウィンドウデザイナのツールボックス

こんなプログラムも書ける

こ，これはシャーペンの……

PICTデータは，受信側のアイテムのresizeプロパティによって，拡大/縮小表示されるかどうかが決まります。

マルチフォントテキストに対応

たとえば，シャーペン上で，上の写真のように入力し，SX-BASICへカット&ペーストし，実行させると写真のように実行されます。

ただし，現在，マルチフォントテキストに対応しているのは，print文，clipboard.str@への代入文だけです。

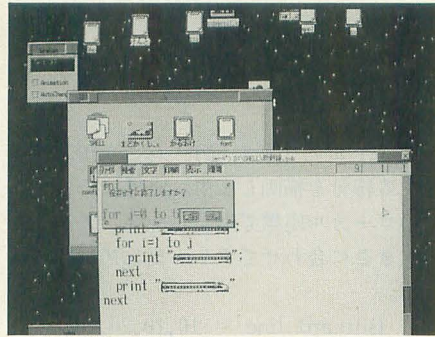
使用できる演算子は「＋」のみです。現在のところSX-BASICの組み込み関数では，マルチフォントテキストを引数としてとれるものはまったくありません。また，結果を収めるべき変数の型もいまのところありません。

クリップボードに対応

SX-BASICでは他タスクとの連絡は，主に独自フォーマットのタスク間通信により行ってきました。このタスク間通信はメッセージフォーマットが独自なために対応アプリケーションは，フリーソフトを中心に何本かあるだけで，大型アプリケーションでSX-BASICと通信可能なものは皆無でした。

SX-BASICで提唱されたタスク間通信のほかにも，アプリケーション間でデータを転送する方法として，クリップボードがあります。

市販ソフトを含め多くのアプリケーションがクリップボードに対応しています。SX-BASICからクリップボードのデータを使えるようになれば，それだけ用途は広がるのです。

クリップボードというと，「シャーペン」などで，カット&ペーストを行う際に用いる文字列の退避領域くらいに思っている人も多いかもしれません。しかし，SX-WINDOW ver.3.0以降では，クリップボードを経由して文字列以外のさまざまなデータが送れます。

「シャーペン」で扱う文章には，ビットマップデータやドローデータなどさまざまな文字列以外のデータが張りつけられます。「シャーペン」自身には，絵を描いたりする機能はないのに，これらのデータを扱えるのは，クリップボードを経由してEasydrawや「パターンエディタ」からデータを転送できるからです。

SX-BASIC ver.0.5から，文字列とスクリプトデータをクリップボードとやりとりできるようになりました。

クリップボードへ文字列を送るには，

```
clipboard.strn = "Hello World !!"
```

のように行います。

逆に，クリップボードの文字列を参照するには，

```
print "クリップボードの内容は";
clipboard.strn;"です"
```

のような操作を行います。

文字列の送信は，従来の装飾なし文字列，マルチフォントテキストの両方を使うことができます。両者の区別はSX-BASICが自動的に行います。

受信の際は，clipboard.strnで装飾なし文字列，clipboard.str@でマルチフォントテキストを受け取ることができます。

スクリプトデータの転送は，

```
clipboard.pict = Bitmap1.script
```

受け取りは，

```
Bitmap1.script = clipboard.pict
```

のように行われます。

一応，転送/受け取りともに，整数値であれば文法的にエラーは発生しないことになっていますが，ここで扱われる値はハンドルですので，足し算や掛け算などを行うと，意味がなくなってしまいます。SX-WINDOW全般にいえることですが，意味のないハンドルを無理に扱おうとすると，システムエラーが発生しますので，取り扱いには気をつけてください。

カレントリソースの変更

これまではSX-BASICで扱うリソースはプログラム名の拡張子をLBに変えたもの1種類しか扱えませんでした。

●openres("ファイル名")

SX-BASICにおけるカレントリソースを変更します。

コードリソース

リソースタイプ「CODE」に収められた，コードリソースを実行することができるようになりました。

codersc(式1，式；サイズ ……)

式1（INT型）で表されるリソースIDを持つ，コードリソースを実行します。コードリソースに制御を移す際に，スタック上にはcoderscで指定されたものを，レジスタにはset_reg()関数で設定された値が代入されます。

スタックに積むパラメータのサイズは以下の3種類があります。

;L　ロングサイズ
;W　ワードサイズ
;B　バイトサイズ

コードリソースの実行が終わった時点でのD0レジスタの値が戻り値として返されます。

例）? codersc(200，123；l)

この呼び出しに対して，

```
#include <sxlib.h>
int
CODEMain(int i)
{
  charbuff[100];
  sprintf(buff,
  "%dが渡されました", i);
  DMError(1, buff);
  /* パラメータを10倍して返す */
  return(i * 10);
}
```

というようなプログラムがリソースファイル「test.lb」中，リソースタイプ「CODE」，リソースID＝200で収められていた場合，SX-BASIC上で，

```
openres("test.lb")
? codersc(200, 123;l)
```

を実行すると，

「123が渡されました」

と，ダイアログが表示され，SX-BASICのウィンドウ上には，1230（＝123×10）が表示されます。

ウィンドウデザイナでの配列アイテム

「こいのぼりPRO-68K」バージョンで配列アイテムをサポートしました。配列アイテムというのは，同じような役割を持ったアイテムをグループ化し，一括管理する機能です。グループはアイテム名で区別し，グループ内の個々のアイテムはその添字で区別しました。で，その添字なのですが，いままでは作成した順に0，1，2，……と自動的に割り振られていきましたが，今回のバージョンからは，ユーザーが好きな値をつけることができるようになりました。

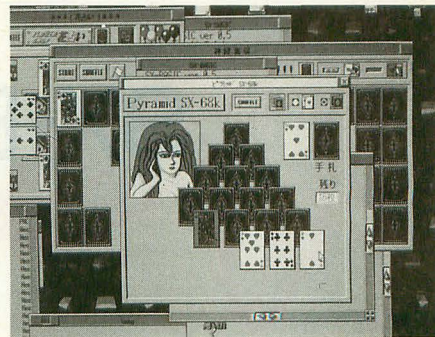
ウィンドウデザイナのツールボックスにはアイテム選択用のアップボタン（数値調整ボタン）があります。ここには現在選択されているアイテムの名前が表示されますが，そのアイテムが配列アイテムだった場合，

アイテム名［添字］

という形式で表示されるようになりました。たとえば，StnBtn1というアイテム名で添字が1だった場合，

StnBtn1[1]

のようになります。ここの「1」の部分をキーボードから変更することにより，好きな値を設定することができます。

ただし，同じプログラムの中に，同じアイテム名と添字を持ったアイテムがあってはいけません。

リソース実行時

SX-BASICはかなり安定してきましたが，新しい機能は新しいバグを生んでいるかもしれません。デバックにご協力ください。

ついにカードゲームが！

まず一般的な注意点です。従来のSX-BASIC用プログラムを新しいバージョンで動かすためには，

▼Window Size (403,190),1,1,sx-win

のようになっている部分を，

▼Window Size (403,190),0,1,1,sx-win

のように変更してください（サイズのあとに0を加える）。

今回の付録ディスクにはSX-BASICに対応したカードゲームが収録されています。これらはすべて共通データリソースとしてCARD.LBを参照しますのでプログラムと同じディレクトリにでも入れておいてください。

これらの作品は郡茂樹さんの投稿作品を編集部で修正したものです。触ってみればだいたいわかるとは思いますが，一応ルールの解説をしておきましょう。

●関西.SXB

漢字の書いてある牌を同じ字のものを2つずつクリックして取り除きます（色は関係ありません）。いまさらいうまでもありませんが，すべての牌を取り除くことがゲームの目的です。

●花占い.SXB

場に出ているカードから同じ数値のものを2枚ずつ取り除いていきます。取れなくなったら台札を1枚ずつめくっていってください。

●神経衰弱.SXB

いわゆる神経衰弱です。カードをシャッフルして場に並べ，適当に2枚ずつめくっていきます。同じ数字なら場から取り除かれます。すべてのカードを取り除くのが目的です。

●ピラミッド.SXB

ピラミッド状に並べられた場札のいちばん下のカード群から2枚のカードを，合計数値が13になるように選択して取り除いていきます。取れるものがなくなったら台札をクリックしてください。すべてのカードを取り除くのが目的です。

リソースの切り換え

さて，本来SX-BASICのプログラムでは1種類のリソースファイルしか扱えませんでしたが，今回のバージョンからはリソースファイルを切り換えて使用できるようになりました。

SX-WINDOWではリソースを扱うとき，リソースタイプとリソースIDというものを使用します。リソースタイプとは“PAT4”であったり“CODE”であったり“MENU”であったりするわけで，それぞれでタイプの何番目かという指定でリソースを特定するわけです。しかし，複数のリソースを扱う場合は，同じタイプで同じIDのものがかちあう可能性も出てきます。

それを避けるためにMacintoshやSX-WINDOWでは「カレントリソース」という考え方を基本にしています。リソースの指定は現在選択しているリソースファイルでなされるわけです。

いかにもシングルタスクのMacintoshらしい考え方ですが，マルチタスク環境でこの考え方を採用すると非常にうっとうしいことになります。複数のリソースファイルを扱う場合は，リソースを参照するたびにカレントリソースの切り換えを行わなければならなくなるからです。「それなら使うものをひとつにまとめておけばいいじゃないか」という考え方もありますが（そもそもはそういうことが推奨されているのでしょう），それではカードゲームなどを作成する場合，7並べとババ抜きとポーカーでほとんど同じような内容のリソースファイルを作成しなければなりません。さらにそれらを同時に起動したときは，同じデータが別々のリソースとしてメモリ上に展開されるといった愚かしいことになります。

実際，かなり使いにくい仕様なのですが，ディスクに収録されたカードゲームでは非共通部をすべて先に表示しておくことで，共通部分を分離しています。

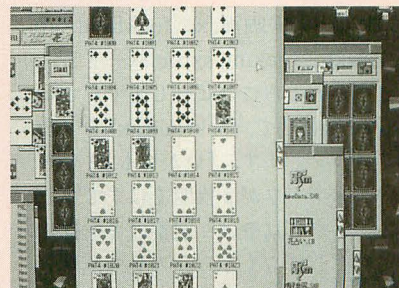
具体的にいうと，非共通部分はウィンドウ作成時に，いわゆる，

▼

で指定しておき，

Bitmap.id＝

というかたちで参照するのは共通部だけに限定しています。これで起動後にカレントリソースを切り換えておけば，いちいちカレントリソースの心配をしなくてもすむわけです。

その分，非共通部の表示制御では姑息な技を使う必要が出てくるのですが，それは具体的にリストを見てもらったほうがわかりやすいかもしれません。

大きく分けて，

.visibleによる表示制御
.modeによる表示制御
.moveによる表示座標制御

の3種類があります。

ボタンのアニメーションなどでは.visibleでも.modeでも対処できます・

ただし気をつけなければならないのは.visible＝0になっているとクリックイベントを拾わないということです。以前のバージョンでは拾っていたのですが，今回のバージョンからは拾わなくなりました。

そこに表示されているものは問わずに，その位置でイベントだけ拾いたいという場合は.modeを使用して.visibile＝1のまま表示を消してください。

ゲームのクリア時にグラフィックを出したいとかいう場合は.moveによる制御がよいでしょう。あらかじめマイナス座標あたりに読み込んでおいたものをウィンドウ内に呼び出すわけです（あまりに絵が大きい場合は.fileのほうが無難でしょう）。

＊　＊　＊

今後カードゲームを作成される方はCARD.LBを使用してみてください。また，カードデータのバリエーションを作る場合などはできるだけこの配列に準拠してください。また，追加データ案がある方は別途ご一報ください。

同様な手法で麻雀ゲームや花札も共通リソースで実現できるでしょう。

しかし，やれば結構いろいろなことができるものです。新たにゲームキットなどもできましたし，SX-BASICプログラマの皆さんの元気な投稿をお待ちしています。

RPGを作ろう

SX-BASIC用ゲーム作成キット

Tamura Kento 田村 健人

SX-BASICを使えば簡単にゲームを作ることはできますが，速度的な問題があります
そこでゲーム専用のエンジンを作成してみました
ウィザードリィ型，ウルティマ型のゲームが作成できます

とりあえず体験してみよう

付録ディスクを展開した中から，sxbasic.x,フィールド.x，フィールド.lb,maze3.x，maze3.lb，文字画面.x，fockfield.sxb，fockmaze.sxb，fockmg.sxbをハードディスクの適当なところにコピーしてください。sxbasic.xを起動して，fockfield.sxbをドラッグ&ドロップします。「いかにも」といった感じのウィンドウが2枚開くはずです。このウィンドウではやはり定番のキー操作ができます。しばらくうろうろして，飽きてください。そしてこう思ってください。「な～んだ。自分ならちゃんと作るのにな」

なにができるのか

フィールド.x，maze3.xは単体でも起動できるので，ダブルクリックしてみてください。

フィールド.x,maze3.xはマップの表示および移動を行うタスクです。特定の地点で指定されたメッセージを親タスク[*1]に送ることができます（このメッセージのことを，以下「地縛事象」と呼びます)。また，編集モードを持っており，マップの作成，地縛事象の登録を行うことができます。

文字画面.xはBASICでいう「locate&print」の感覚で文字表示を行うためのタスクです。

これらをSX-BASICからタスク間通信で制御して，RPGでも作ってしまおうというのが目的です。

＊1　親タスク:タスクを起動したタスクのこと。ディレクトリ表示ウィンドウからダブルクリックして起動した場合，親はディレクトリ表示のタスクで，sxbasic.xからfock()で起動した場合，親はsxbasic.xとなる。

マップを作る

フィールド,maze3ともに，マップの横サイズをx，縦サイズをyとするとマス目には0～x×y-1の番号がついています。マップの左上が0番で右側に1,2と続きます。0番のマス目の下はx番となります。マップ上の位置は，このマス目の番号で表します。

●フィールド

まずは下準備です。アイコンリスト.Xを起動して，その中の＊.lbをダブルクリックします。アイコンメンテ.Xが起動するはずです。そこで，

　実行ファイル:　　パターン一覧.X
　実行オプション:　-o%

と書き換えて登録ボタンを押します。

アイコンメンテ.Xとアイコンリスト.Xを終了させて，Xアイコンのメニューから「再起動」を選びます。「ファイルに保存しますか？」というようなダイアログが出るので「はい」のほうを押してください。以上の作業で，＊.lbのアイコンをダブルクリックするとパターン一覧が起動するようになります。

では，フィールド.lbのアイコンをダブルクリックしてパターン一覧を起動してください。ここで，マップのパーツ（以下，地図素片)を横32縦，32ドットのPAT4で作成します。既存のものを変更しても，新規に追加しても構いません。ただし，リソースIDに隙間を空けないようにしてください。リソースIDというのは，パターン一覧で各パターンの下に書かれている数値のことです。パターンの削除を行わない限りリソースIDは連続になるはずですから，削除は行わないようにすればいいでしょう。地図素片はリソースIDが128～32767（16進数なら0x0080～0x7fff）の範囲で作成できます。

パターン一覧は終了するときにセーブするかどうか聞いてきます。保存したいときはここで「はい」のボタンを押します。

マップの新規作成はSX-WINDOW上ではできません。コマンドシェルかシャーペンのコンソールで行うことになります。作業には付録ディスクに入っている mknullmap.xとarlk.xを使います。マップはリソースタイプFMAPで，IDは地図素片と同じく128～32767です。

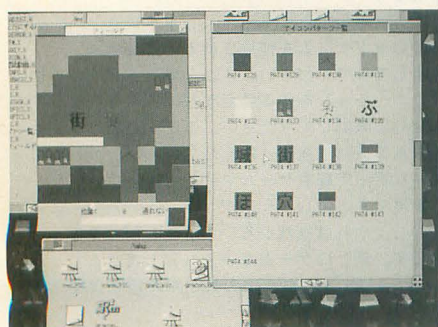
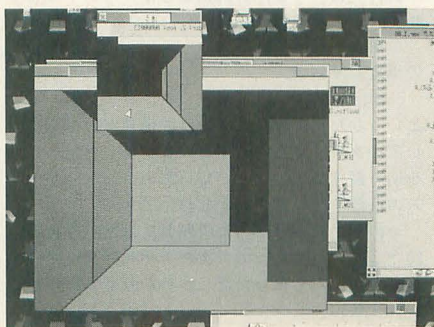
横256，縦200のマップを作りたいときは，

```
A>mknullmap 256 200 FMAP.rsc
```

とします。マップの大きさは，縦横ともに32767ぐらいまでで，メモリに読み込める程度でなければなりません。最後のファイル名はFで始まればなんでも構いません。このマップが1枚目ならば，

```
A>arlk  -a  フィールド.lb FMAP0080  -o FMAP.rsc
```

としてフィールド.lbに入れてやります。

FMAP0080は必ず大文字にします。もし2枚目なら，FMAP0080のところをFMAP0081とします。後ろ4文字が16進数のリ

パターン一覧でリソースを見る

マップを作っていく

ソースIDとなっているわけです。すでにあるIDを指定すると上書きされます。地図素片と同じく，リソースIDに隙間があってはいけないことになっているので，IDの指定には気をつけてください。

A＞arlk -l フィールド.lb FMAP

でFMAPの一覧が見られるのでIDが連続していることを確認してください。なお，シャーペンのコンソールでこれらの作業を行うときは，フィールド.lbがオープンされていないことを確認し，作業中はほかになにも行わないようにしてください。

マップの中身を実際に編集するのはSX-WINDOW上で行います。フィールド.xを起動して，Eキーを押すと編集モードになります。編集モードでのキー操作は表1を見てください。

地図素片と通行可能性（通れるかどうかの属性）は1対1の対応となっています。同じ見かけで通行可能性が異なる地図素片を作りたいときは，同じパターンを2つ登録します。

地縛事象は27バイトまでです。

●maze3

マップの新規作成はやはりコマンドシェルで行わなければなりません。リソースタイプがdmapとなるので，

A＞mknullmap 256 200 dmap.rsc

A＞arlk -a maze3.lb dmap0080 -o dmap.rsc

のようにします。dmap0080は必ず小文字です。リソースIDが連続していなければならないのもフィールド.xと同じです。

マップの中身の編集はSX-WINDOW上でできます。maze3.xを起動してEキーを押すことで編集モードになります。キー操作は表2を見てください。

maze3.xでは1枚の壁で表と裏が別のデータになっています。視点から見た面に壁がなければ描画されないので，一方からは見えるが逆からは見えない壁を作ることができます。編集モードでは普通は両面から見える壁を作りますが，Tキーを押すことにより見える側の面にしか壁を作らなくなります（図1）。

通常，なにもないところと特定の色の壁のみが通行可能です。Fキーを押すと目の前の壁の通行可能性が通常と反転します。この反転動作も，Tキーによる片面/両面切り替えの影響を受けます。

タスク間通信

SX-BASIC形式のメッセージによりさまざまな制御を行います。

●フィールド（受信）

・CMAP <map-id>, <loc>

リソースIDが <map-id> のマップに切り替え，主人公を <loc> の位置へ移動させる。

・CLOC <loc>

現在のマップ内で，主人公を <loc> へ移動させる。

・CPAT <loc>, <pat-id>

現在のマップの <loc> の位置をリソースID <pat-id> のパターンに書き換える。

・？MAP

現在マップ問い合わせ。“MAP <map-id> ”が返る。

・？LOC

現在位置問い合わせ。“LOC <loc>”が返る。

・？PAT

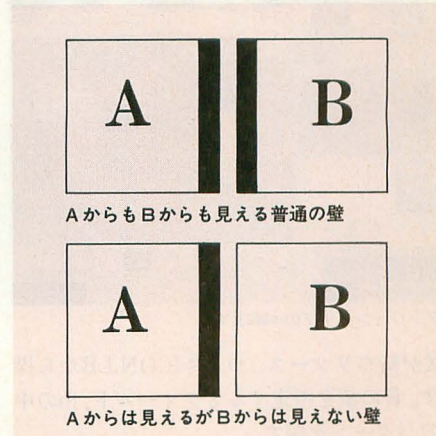
現在位置の地図素片問い合わせ。“PAT <pat-id> ”が返る。

・？PAT <loc>

現在マップの <loc> の位置の地図素片を問い合わせる。“PAT <pat-id> ”が返る。

・LOCK <taskid>

主人公をいっさい移動できないようにし，キー入力を <taskid> のタスクへ横流しする。指定するタスクはキー入力の横流しに対応するものでなければならない。現在はフィールド.x,maze3.x，文字画面.xだけが対応している。

・UNLOCK

LOCKによる移動制限を解除する。

・ELOCK

Eキーによる編集モードへの移行を禁止。

・EUNLOCK

Eキーによる編集モードへの移行を許可。

・PROB <n>

マップ移動中，<n>/32768の確率で親タスクに“ENCOUNT <loc> ”を返す。

・PUTPAT <n>, <x>, <y>, <task>, <type>, <id>, <mode>

表示画面中の指定位置に指定パターンを描画する。

<n>:セル番号（0～9）。スプライト番号のようなものだと思えばよい。画面中には10枚のパターンしか表示できない。0番がもっとも奥，9番がもっとも手前に表示される。

<x>, <y>:座標（それぞれ0～16）。ドットで指定するのではなく，表示画面中の割合。画面の左上が0，0で，右下が16，16。指定座標を中心としてパターンを描画する。

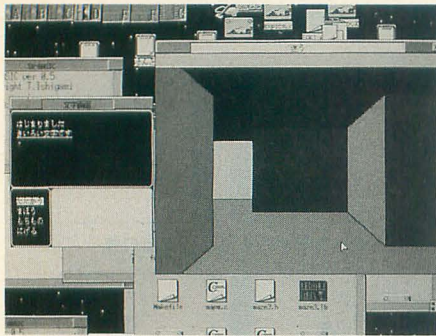
<task>:パターンを探すリソースを保持するタスクのID。0を指定するとSXWIN.

図1 maze3 データ概念図

表1 フィールド.x 編集モードでのキー操作

キー	操作
[SPACE]	現在位置に地図素片を置く
[RUP]/[RDOWN]	置く地図素片を切り替える
[T]	現在の地図素片の通行可能性を反転する
[SHIFT]+[RUP]/ [SHIFT]+[RDOWN]	マップを切り替える
[改行]	地縛事象を入力する
[S]	編集結果をフィールド.lbに保存する
[E]	編集モードから抜ける

表2 maze3.x 編集モードでのキー操作

キー	操作
[SPACE] [OPT.1]+[↑]	現在の色の壁を目の前に置く
[OPT.1]+[←]	現在の色の壁を左側に置く
[OPT.1]+[→]	現在の色の壁を右側に置く
[OPT.1]+[↓]	現在の色の壁を後ろに置く
[D]	目の前の壁を消す
[F]	目の前の壁の通行可能性を反転する
[RUP]/[RDOWN]	壁の色の切り替え 白→明灰→暗灰→黒→黄→赤→緑→青→白
[SHIFT]+[RUP]/ [SHIFT]+[RDOWN]	マップを切り替える
[T]	壁を置くときに，向こうの面には置かない もういちど押すと，両面に同じ壁を置く状態に戻る
[改行]	地縛事象を入力する
[S]	編集結果をmaze3.lbに保存する
[E]	編集モードから抜ける

ダンジョンタイプのmaze3.x

Xが持つリソース，つまりICON.LBから探す。負の値を指定するとフィールド.lbの中のパターンを探す。

〈type〉:3 か 4。3 が PAT3，4が PAT4。

〈id〉:リソースID。

〈mode〉:描画モード。

0: 標準
1: 反転
2: ハイライト
3: ハイライト反転
4: 消去
6: 網掛け
7: 網掛け反転
8: 不可視
9: 不可視反転

セルのパターンを変更するとき，特にDELPATをする必要はない。同じセルに対してPUTPATを連呼して構わない。

・DELPAT　〈n〉
〈n〉番のセルに置いたパターンを消去する。

・DELPAT ALL
0～9番のセルのパターンを消去する。

・REDRAW
画面を再描画する。

・ACTIVATE
ウィンドウをアクティブにする。

●フィールド（送信）

以下のSX-BASICメッセージを発行します。宛先は，問い合わせに対する答えのときは問い合わせたタスクに，それ以外ではフィールド.xを起動したタスクです。

複数のメッセージが';'で区切られていっぺんに送信されることがあります（例：”CAREA 128 ; ENCOUNT 439”）。

・MAP　〈map-id〉
現在マップ問い合わせの返答。

・LOC　〈loc〉
位置問い合わせの返答。

・PAT　〈pat-id〉
地図素片問い合わせの返答。

・CAREA　〈pat-id〉
移動によって主人公直下の地図素片が変わったとき，変わったあとの地図素片の種類を知らせる。

・HIT　〈loc〉
〈loc〉にある通れない地図素片にぶつかった。

・ENCOUNT　〈loc〉
ある確率で発行。

・決
テンキーの5，改行キー，スペースのいずれかが押された。

・消
テンキーの0，ESCキーが押された。

・その他（最大27バイト）
地縛事象。編集モードで設定した位置にくると発行。

●maze3・受信

〈dir〉は0:左，1:上，2:右，3:下。

・CMAP　〈map-id〉，〈loc〉，〈dir〉
リソースIDが〈map-id〉のマップに切り替え，主人公を〈loc〉の位置，〈dir〉の向きに移動させる。

・CLOC　〈loc〉，〈dir〉
現在のマップ内で，主人公を〈loc〉の位置，〈dir〉の向きに移動させる。

・CPAT　〈loc〉，〈dir〉，〈wall〉
現在のマップの〈loc〉の〈dir〉の向きの壁を〈wall〉に書き換える。〈wall〉は0が壁がない状態，8～15が壁の色。128を加えると通行可能性が反転する。

・？LOC
現在位置問い合わせ。”LOC 〈loc〉，〈dir〉”が返る。

・SKY　〈c〉
空の色を〈c〉にする。画面の再描画はしないので“REDRAW”を呼ぶ必要がある。色は以下のとおり。

8　白
9　明灰
10　暗灰
11　黒
12　黄
13　赤
14　緑
15　青

・EARTH　〈c〉
地面の色を〈c〉にする。再描画しない。

・DOOR　〈c〉
〈c〉の色の壁を通行可能にする。0を指定するとデフォルトと同じで，壁がないところのみ通行可能。

・LOCK　〈taskid〉
・UNLOCK
・ELOCK
・EUNLOCK
・PROB　〈n〉
・？MAP
・PUTPAT　〈n〉，〈x〉，〈y〉，〈task〉，〈type〉，〈id〉，〈mode〉
・DELPAT　〈n〉
・DELPAT ALL
・REDRAW
・ACTIVATE
以上はフィールド.xと同じ。

●maze3・送信

宛先に関してはフィールド.xの送信と同じです。複数のメッセージが';'で区切られて送られることがあります。

・LOC　〈loc〉，〈dir〉
位置問い合わせの返答。

・HIT　〈loc〉，〈dir〉
〈loc〉の〈dir〉の向きにある通れない壁にぶつかった。

・MAP　〈map-id〉
・ENCOUNT　〈loc〉
・決
・消
・その他（最大27バイト）
以上はフィールド.xと同じ。

文字画面

フィールド.xとmaze3.xには文字を表示する機能がありません。文字の表示専用のタスクが文字画面.xです。

文字画面.xはウィンドウ内に「文字窓」という領域を0番から9番まで10枚開くことができます。それぞれの文字窓が文字の色・背景の色・枠の色・座標系を持っています。「文字窓に文字列を描く」というのが文字画面.xの主な仕事です。

文字画面.xが起動された直後には，ウィンドウいっぱいに0番の文字窓が開かれています。普通の文字窓では，文字窓の横幅をはみ出す文字列を描画するとそのままはみ出して描画してしまいます。0番の文字窓は特別に行の折り返しとスクロールをサポートしています。以下，0番の文字窓をスクロール付き文字窓とも表記します。

0番以外の文字窓を開きたいときはメッセージADDLINE/SETLINEを用いてウィンドウの行数を広げます。広げることによって0番の文字窓に属さない領域ができるので，そこにメッセージOPENで文字窓を開きます。

文字窓は任意の順序で開閉できるわけではありません。メッセージCLOSEで閉じるのは最後に開いた文字窓です。これは文字

窓の管理がスタック形式になっているためです。

●文字画面.xの起動

以下のようなコマンドラインを必要とします。コマンドラインになにも指定しない場合にはそれなりに起動します。

"文字画面.x －k 〈id0〉,〈id1〉,〈dx〉,〈miny〉,〈maxy〉,〈win〉,〈fore〉,〈back〉,〈border〉"

〈id0〉には通常時にキー入力の横流しを行う先をタスクIDで指定します。フィールド.xかmaze3.xのタスクIDとなると思います。

〈id1〉は,後述するメッセージSTARTKEYとENDKEYの間にキー入力の解釈を行うタスクのIDを指定します。普通はsxbasic.xなど,全体を管理するタスクです。

－k 〈id0〉,〈id1〉オプションがない場合,キー入力に関してはなにも行いません。

〈dx〉,〈miny〉で初期のウィンドウの大きさを指定します。横幅〈dx〉文字,縦〈miny〉行のウィンドウになります。この指定がそのままスクロール付き文字窓の大きさになります。文字画面.xのウィンドウはここで指定した大きさ以下にはなりません。また,横幅はここで指定した値より変更することができません。

〈maxy〉は,ウィンドウを何行ぶんまで広げられるかを指定します。

〈win〉,〈fore〉,〈back〉,〈border〉はそれぞれ,文字窓がない部分の色,スクロール付き文字窓のデフォルトの文字の色・背景の色・枠の色です。

-f0,-f1,-f2を指定すると,表示が12ドット,16ドット,24ドットフォントになります。

●受信メッセージ

・SMES 〈attr〉,〈str〉

スクロール付き文字窓に対して〈attr〉の属性で文字列〈str〉を書く。改行コードはchr＄(10)。

〈attr〉は8,9,a～f,n,uの組み合わせから成る。

8　白文字
9　明灰色文字
a　暗灰色文字
b　黒文字
c　黄色文字
d　赤文字
e　緑文字
f　青文字
n　反転
u　下線付き

たとえば"en"と指定すると赤色の反転文字となる。" "のように空のときは描画する文字窓のデフォルトの色が用いられる。

・CMES 〈w〉,〈x〉,〈y〉,〈attr〉,〈str〉

〈w〉番の文字窓の〈x〉〈y〉の座標に〈attr〉の属性で文字列〈str〉を書く。改行コードは認められない。文字窓でクリッピングを行うわけではないので,文字数については自己管理すること。文字窓が枠の描画に1文字ぶん使用していることを考慮すること。

・ADDLINE 〈line〉

ウィンドウの行数を〈miny〉+〈line〉にする。〈miny〉+〈line〉が〈maxy〉を超えてはならない。

・SETLINE 〈line〉

ウィンドウの行数を〈line〉にする。〈line〉の値は〈miny〉～〈maxy〉の間である。

・SUBLINE / DELLINE

ウィンドウの行数を〈miny〉にする。

・OPEN 〈x〉,〈y〉,〈dx〉,〈dy〉,〈fore〉,〈back〉,〈border〉

〈x〉〈y〉の座標から大きさ〈dx〉〈dy〉で指定された色の文字窓を開く。最初にこのメッセージで作った文字窓が1番の文字窓となり,あとはこれを呼ぶたびに加算される。文字窓は9番まで。文字窓同士が重なる位置に開くとどうなるのかは作った本人も知らない(単に上書きするだけだと思いますが……)。

・CLEAR 〈w〉

〈w〉番の文字窓の中の文字をすべて消去する。

・CLOSE

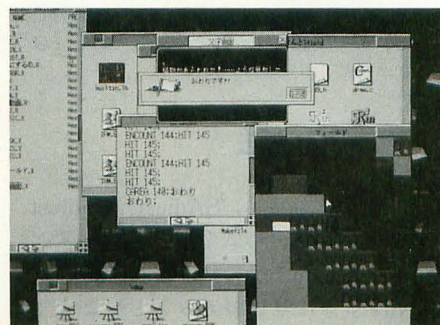
最後に開いた文字窓を消去する。

・PUTCUR 〈w〉,〈x〉,〈y〉

〈w〉番の文字窓の〈x〉〈y〉にウェイトマークを描画する。〈w〉が0のときは〈x〉〈y〉は無視され,カーソル位置に描画する。

・DELCUR

ウェイトマークを消去する。

・STARTKEY / ENDKEY

フィールドタイプのサンプルゲーム(?)

この2つのメッセージの間,文字画面.xに対するキー操作が-k 〈id0〉,〈id1〉オプションで指定した〈id1〉のタスクにSX-BASIC形式で送られる。送信の内容は以下のとおり。

←, 4	"左"
↑, 8	"上"
→, 6	"右"
↓, 2	"下"
SPACE, RET, 5	"決"
ESC, 0	"消"

おまけのjoy10key.r

やっぱりジョイスティックが使えなきゃね,ということでジョイスティックが操作されるとそれに対応するテンキーが押されたことになるソフトです。Aボタンは5キー,Bボタンは0キーとしています。スタートアップメンテに放り込んで再起動すると使えるようになります。

プログラミング

この記事をひととおり読んだところで,なにをすればいいのかはわからないでしょう。これはあなたの理解力の問題ではなくて,私の文章力の問題です。とにかくfockfield.sxbとfockmg.sxbを読んでください。

重要な部分はすべてSX-BASICでコーディングすることになりますが,SX-BASICの特殊な部分にはさほど触れる必要がありません。X-BASICと同じ感覚でプログラムを組めると思います。

フィールド.xやmaze3.xの描画は遅いです。なにもSX-WINDOWでやらないでコマンドライン上でやればもっともっと速くなるでしょう。それでもSX-WINDOWで実現する理由は,

タスク間通信の醍醐味	5%
SX-WINDOWが好きだから	15%
成り行き	80%

です。

時間がなくてまともなサンプルが作れなかったのが心残りです。「バグがない」といえる自信はありません。んー,プログラムのバグよりは仕様のほうが問題ありそうですね。はたしてこのままの仕様でまともなソフトが作れるのでしょーか。

わからないことがあったらアンケートハガキにでも書いて送ってください。機会があればお答えできるでしょう。

SXの表示能力を拡大する

SAdjust.r

Fukushima Shota 福嶋 章太

画面は広くしたけれどもSX-WINDOWがついてこないとお嘆きの方
SAdjust.rですべて解決です
画面を自在に設定するほか，さまざまな機能を備えています

ディスプレイ怠けてませんか?

瀧さんの2回にわたる「メガディスプレイ」の記事*1は読みましたか？（読んでない人は，この記事を読む前に読んでおくことをおすすめします）あの記事で，ディスプレイにはまだまだ隠された底力があることがわかりました。最近，低価格化が進んでいる高性能ディスプレイはいうまでもなく，たとえ純正のモニタであっても，その性能のすべてを出し切っているわけではないのです。

ディスプレイが怠けているのがわかった以上，それを許しておけないのが真のX68000ユーザーってもんです（よね？）。いじれるものはソフトであれハードであれ，とことんいじり倒さねば気がすみませんから，怠けたディスプレイがあるのなら，その性能を限界まで引き出してやり，なおかつそれを有効に活用してあげなくてはなりません。

そこで，ディスプレイの性能を引き出すための手助けになり，同時にシステム（今回はSXシステム）との辻褄合わせも行うプログラム，SAdjust.rを紹介させていただきます。あなたの怠けたディスプレイを，SAdjust.rでたたき直してあげてください。

＊1 瀧さんの「メガディスプレイ」の記事は1994年6月号と9月号に載ってます。

SAdjust.rとは

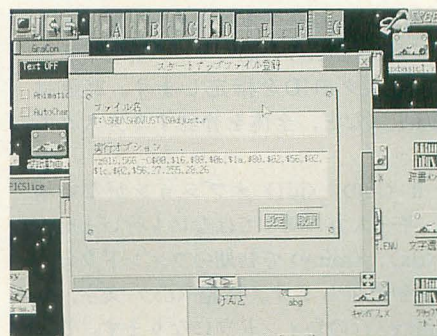
純正のシステム（確かver.2.00あたりから）で，スタートアップメンテ.X*2を起動すると，いくつかあるアイコンの中にadjust.?というアイコンが見えると思います。adjust.?とはシステムリソースであるBUILTIN.LB*3にビルトインコマンド*4として含まれているadjust.rのことで（.?というのはワイルドカード*5），主に表示画面のスクロール機能などを管理するプログラムです。今回発表するSAdjust.rはこのadjust.rからの上位互換の形式をとっています。

「メガディスプレイ」の記事を読むと，ディスプレイの性能を引き出すことは，CRTCレジスタの値をいろいろといじってやることによって実現できるようです。ただし，ディスプレイの種類や同じディスプレイであっても，その個体差によってCRTCに設定できる限界値は違います。そこでSAdjust.rではCRTCの値は，ユーザーが自由に設定できるようにしました（詳しい設定方法は起動スイッチ以降のところで説明します）。

とまあ，adjust.rの機能を継承しつつ，CRTCレジスタの設定を可能にし，それをSXシステム上で矛盾が生じないように辻褄合わせをするというのが，SAdjust.rの主な機能です（そのほかにもSAdjust.rには付加機能がいくつかありますが，それらについても起動スイッチ以降でまとめて説明します）。

＊2 スタートアップメンテ.Xはシステムの起動時に実行されるプログラムを設定するためのもので，システムインストール直後の状態で，adjust.?のほか，sfile.?やhenwin.?などが設定されています。
＊3 BUILTIN.LBのほか，SYSTEM.LB，ICON.LBがシステムリソースに含まれます。
＊4 CODE,CMDSの2種類リソースによってBUILTIN.LBに含まれるプログラムをビルトインコマンドと呼びます。
＊5 .?はプログラムが.Rと.Xのどちらの形式でも実行できるようにするための処置だと思われますが，わかりにくい（アイコンが白紙になる）ので素直に.Rや.Xと指定しておいてほしかったところです。

組み込み方法

すでに書いたように，SAdjust.rはadjust.rからの上位互換の形式をとっています。ですから，SAdjust.rの組み込みはadjust.rをSAdjust.rに差し替えることによって行います。なお，以下の説明はシステムをインストールした直後の状態を前提に説明していますので，環境をカスタマイズされている方は，各自の環境にあわせて組み込みを行ってください。

スタートアップメンテの設定

では具体的に説明します。

とりあえず，SAdjust.rを実行可能な状態にします（SHELLまたはアクセサリディレクトリにでも入れてください）。

次に，スタートアップメンテ.Xを起動してください。ウィンドウが開いたら，その中のadjust.?をダブルクリックし，ファイル名をSAdjust.rに変更します。

次に，メニューメンテ.Xを起動します。ウィンドウが開いたら，メニュータイプをアイコン用に，メニュー番号を0番にしてください（初めからなってる）。

左のメニュー内容の中から「スクロール・オン」を選んでください。選ぶと，実行ファイルがadjust.rになっているはずですから，それをSAdjust.rに変更します。同様に「スクロール・オフ」にも行ってください。

以上の作業で，純正システムにおけるSAdjust.rの組み込みは完了です。環境をカスタマイズしている場合は，このほかにもadjust.rとある場所を片っ端からSAdjust.rに変更する必要があります。

ひとつ注意しなければならないことがあります。adjust.rとSAdjust.rは非常に相性が悪いため，もし同時に起動してしまう

ようなことがあると，表示画面が乱れるなどの現象を引き起こしてしまいます。そうなってしまった場合はどちらかを終了させなければならないので，起動中のタスクをなるべく少なくしたあと，adjust.rかSAdjust.rのうち，あとから起動したほうを-Rスイッチつきで実行してください。

起動スイッチ

まずは，純正adjust.rにも備わっている起動スイッチから説明します。斜体文字の部分には，10進の数字か，$をつけた16進の数字を指定します。

-Xx,y

実画面上における表示位置の設定をします。x，yに表示画面の左上角の絶対座標*6を指定します。

-Sn

スクロールのON/OFFを設定します。n＝0でON，n＝1でOFF，またn＝−1でトグル動作（状態の反転をする）をします。

-R

SAdjust.rを終了しますが，場合によっては終了できないこともあります（理由はadjust.r解析を参照してください）。

と，ここまでが純正adjust.rにも備わっているスイッチです。

次に，SAdjust.r独自の起動スイッチを説明します。

-Mx,y

マウスカーソルの初期座標を設定します。x,yに絶対座標で指定します。

-Cn0,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15

CRTCレジスタの値を設定します。このスイッチがSAdjust.rの目玉です。この設定をいろいろいじくって，ディスプレイをいじめてあげましょう。n0〜n15の値に関しては表1を見てください。

-Zx,y

画面のスクロールサイズを設定します。x,yにスクロールサイズを指定します。このスイッチは基本的に設定する必要はありません。通常スクロールサイズは，-Cスイッチで指定した値から計算される表示サイズと同じ値が設定されるので，表示サイズとは違う値を設定したいときのみ使用します。

-Nx,y

スクロールマージンサイズ（マウスカーソルに追従して画面のスクロールを開始する，画面端からの幅）を設定します。x,yにそれぞれ横と縦のマージンサイズを指定します。

-Kn

SYSDTOP.SXに状態を保存するかどうかの設定をします。n＝0で記憶する，n＝1で記憶しない，またn＝-1でトグル動作をします。

-Tn

テレビコントロールを行います。当然テレビコントロールケーブルがつながるディスプレイでしか動作しません。nの値に関しては表2を見てください。

-Un

終了時のCRTC値復帰モードの設定をします。n＝0で復帰ON，n＝1で復帰OFF，またn＝-1でトグル動作をします。復帰OFFにした場合，CRTCの設定値によっては画面が乱れる可能性があります。

以上がSAdjust.rの全起動スイッチです。

＊6　実画面（デスクトップ）上での座標です。

SX-BASICからのメッセージ

SAdjust.rでは以下のメッセージをSX-BASIC形式で受け取ります。

HOME x,y

-X スイッチと同様の動作をします。

SCROLL n

-S スイッチと同様の動作をします。

QUIT

-R スイッチと同様の動作をします。

END

-R スイッチと同様の動作をします。

MOUSE x,y

-M スイッチと同様の動作をします。

CRTC n0,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15

-C スイッチと同様の動作をします。

SIZE x,y

-Z スイッチと同様の動作をします。

MARGIN x,y

-N スイッチと同様の動作をします。

STARTUP n

-K スイッチと同様の動作をします。

CRESET

SAdjust.rを通さずに画面モードやCRTC値を変更した場合に，値をSAdjust.rの設定値に戻します。

VERSION

SAdjust.rのバージョンをSX-BASICのメッセージ形式で返します（SADJUST VER.1.29）。

GET SCROLL

スクロールON/OFFの設定値を返します（SADJUST SCROLL n）。

GET CRTC

CRTC の設定値を返します（SADJUST CRTC n0,n1,n2,…）。

GET SIZE

スクロールサイズの設定値を返します（SADJUST SIZE x,y）。

GET MARGIN

スクロールマージンサイズの設定値を返します（SADJUST MARGIN x,y）。

以上が SAdjust.rが受けつけるSX-BASICのメッセージです。

設定例

では，とりあえず先月号の45ページに瀧さんの調べた「モニタ別CRTC設定表」というのが載っていますので，そのうちの「1)ノンインタの広い1：1モード」とい

表1　CRT設定値
（値はすべてバイト範囲で指定）

n0	システムポート#4(bit1のみ有効)
n1	CRTC R20 (low)
n2	CRTC R00
n3	CRTC R01
n4	CRTC R02
n5	CRTC R03
n6	CRTC R04 (high)
n7	CRTC R04 (low)
n8	CRTC R05
n9	CRTC R06
n10	CRTC R07 (high)
n11	CRTC R07 (low)
n12	CRTC R08
n13	Sprite CTRL R0
n14	Sprite CTRL R2
n15	Sprite CTRL R3

表2　テレビコントロール
nの値により以下の動作をします

$01/$02/$03	ボリュームを上げる/下げる/普通にする
$04	チャンネルコール
$05	テレビ画面(初期化、リセット)
$06	音声ミュート
$07/$0D	電源 ON/OFF
$08	テレビ/コンピュータ
$09	テレビ/外部、コンピュータノーマル/オーバー
$0A	コントラストノーマル
$0B/$0C	チャンネルアップ/ダウン
$0E	電源 ON/OFF切り換え
$0F	スーパー1
$10〜$1B	チャンネル1〜12
$1C	テレビ画面($05)
$1D	コンピュータ画面($05+$08)
$1E	スーパー1($05+$0F)
$1F	スーパー2($05+$0F+$0A)
$20+上記	電源をONして上記のファンクションを実行する

うのを試してみましょう。やることは，レジスタ設定値というのと表1を対応させて，-Cスイッチを設定するだけです。

まずn0というのはHRLのことで，この場合HRL＝0ですので$00を指定します。もし，HRL＝1にしたい場合には$02を指定します（ビット1を立てるから）。

次に，n1～n5は特に難しく考える必要はなく，そのまま対応させて$16,$8a,$0f,$17,$80と指定します。

次に，n6,n7ですが，これにはR04の値を上位と下位に分けて指定します。この場合R04＝$0297ですので$02,$97と指定します。

同じように対応させて，n8～n11には$00,$21,$02,$97と指定します。

次のn12には，通常$1bを指定します（特に値をいじる必要はありません）。

最後の，n13～n15ですが，これはスプライト用の設定です。SXの場合，通常スプライトは表示しませんので，ここの設定は特に重要ではありません。念のため，n13に$ff，n14にn9と同様の値，n15にn1と同様の値を設定してください。

以上のように対応させた結果，

-C$00,$16,$8a,$0f,$17,$80,$02,$97,$00,$21,$02,$97,$1b,$ff,$21,$16

という設定ができあがります。

最後に

SAdjust.rは，あくまでディスプレイの性能を引き出す手助けをするプログラムにすぎません。あとは，瀧さんの記事やディスクに付属のドキュメントを参考に，あなたが，あなたのディスプレイの性能を極限まで引き出してあげてください。

「次にCRTCをいぢめるのはあなたです」

ん～，かっこいいなぁ。

SAdjust.rを作るにあたって，シャープ純正のadjust.rを大幅に参考にさせていただきました。制作者の方々に厚くお礼申し上げます。

adjust.r解析

adjust.rを解析していてわかったことをいくつか挙げてみます。ちなみにSX用のプログラムを組んだことのある人を対象にしています。

それから，これは私が独自に解析したものなので，以下のことに関してシャープなどに問い合わせることはしないでください。

●タスクマンイベントコード1（ENDTSK）の引数2の意味

このイベントはシステムが終了時や再起動時にタスクIDの若い順に発行する，各タスクに終了を通知するイベントです。すべてのタスクにイベントを発行したあとに，終了せずに残っているタスクがある場合は再びイベント発行を行い，すべてのタスクが終了するまで繰り返しています（どうしても終了しないタスクがいると，255回で諦めます）。

そこで引数2ですが，どうやら［イベントを発行した回数÷2＋割り算の余り］という値が入ってくるようです。

adjust.rでは1回目のイベントを無視し，2回目以降のイベントで終了処理をしています（引数2が$00000000以外のときに終了処理を行う）。ちなみにadjust.rがすでに動作している状態でadjust.r -rのようにしてadjust.rを終了させる場合，adjust.rは引数2に$00000080を入れてENDTSKイベントを発行しています。

●ベクタをフックするプログラムの終了方法

コールベクタをフックするプログラムを終了させる場合，当然ベクタを元に戻すわけですが，ただ元のベクタをセットして終了するだけでは同じコールをフックしているプログラムが複数起動している場合に不具合が生じる可能性があります。

図1を見てもらえばわかると思いますが，起動をした順番とは逆の順番で終了をする場合には特に問題はありません。ですが，それ以外の順番で終了すると，終了後のベクタがフリーブロック内を指すことになり，最悪ハングアップということにもなります。

ですから，同一のコールベクタをフックするプログラムの終了は，起動とはまったく逆の順番で行う必要があるわけです。

実際それを実現するためにadjust.rの中では，終了時にベクタが自分で設定したものと異なっていた場合，終了をしないという処理をしています。

この方法だと終了イベントなどを無視することにもなりますが（adjust.r -rを実行しても終了しないときはこれが原因です），純正アプリの仕様ですからコールベクタをフックするプログラムを作る場合は，この仕様に右へ倣えすることをおすすめします（開発キットのプログラマーズマニュアルにもちょこっと載ってます）。

ただし，いくらadjust.r（SAdjust.r）がこのような仕様になっていても，ほかのプログラムがこのようになっていなければ，やはり不具合が発生します。ですからSAdjust.rでは，このような不具合が発生しそうな場合には，ダイアログでメッセージを表示するようにしておきました。

●垂直同期割り込みのバグ

adjust.rは垂直同期割り込みを利用して画面のスクロールを行っています。これによって，実画面モードであってもマウスカーソルが常に画面内にあるようになっています。

と，書きたいところですが，実は1カ所だけマウスカーソルが表示されなくなるところがあります。それは座標(0,0)です。

試しに，実画面の右下のほうを画面に表示させた状態からマウスカーソルを勢いよく左上に移動させてみてください。

どうでしょうか。マウスカーソルが座標(0,0)にあるにもかかわらず，画面には(0,0)が表示されていないはずです。

たいしたことではありませんが，私はこれが前々から気になっていたのでSAdjust.rではこのバグを直しておきました。

●ハンドルロックのフック

adjust.rでは，およそ無関係と思われるハンドルロック系のコール（__MMHLockと__MMHdlLock）をフックしています。実際解析してみると，本来のadjust.rの機能とはまったく関係ない処理をしていました。では，なにをしているのでしょうか？

68000系のCPUには68020からオンチップキャッシュというものがついています。adjust.rでフックしているルーチンではハンドルロックの処理のあと，このオンチップキャッシュをクリアしています。

念のためFSX.X内の本来のコールベクタのほうも調べてみましたが，やはりオンチップキャッシュの制御はしていませんでした。

とまあ，ここまで書けばおわかりとは思いますが，X68030でオンチップキャッシュをONにしてSX-WINDOWを使う場合は，必ずadjust.r（もしくはSAdjust.r）を起動するようにしたほうがよさそうです。

しかし実はこの処理，ver.3.1付属のadjust.rではしっかり取り払われていました（システムでちゃんと処理するようになったのでしょう）。というわけで，SAdjust.rではSXのバージョンにより処理を振り分けています。

図1　ベクタフックの様子

エディット環境パワーアップ

シャーペン用外部コマンド

Tamura Kento 田村 健人

外部コマンドで自在に拡張できるようになったシャーペン
その気になれば，それ自体をシェルのように扱うこともできます
シャーペンを拡張することで広がる可能性を垣間見てみましょう

先月号の私の記事中では「残り6個をディスクに収録」ということでしたが，その後4個増え，さらに先月号のtohan.exに不具合があったのでそれも収録してもらいました。

付録ディスクを展開した中から，*.exのファイルをシャーペンのEXCOMディレクトリにコピーしてください。EXCOMディレクトリはシャーペン.Xがあるディレクトリにあるはずです（A:/シャーペン/EXCOMなどの位置になっていると思います)。あとはキーに定義するなり煮るなり揚げるなりしてください。

キー定義ファイルの編集

まだシャーペンに慣れていない方のために，キー定義変更の方法を詳しく説明します。シャーペンは「シャーペンモード」,「エディタモード」,「タイプモード」,「コンソールモード」があります。

エディタモードとタイプモードはシャーペンモードで代用できますから，私はエディタモードとタイプモードは使わないで，シャーペンモードのみカスタマイズしています。

まず，シャーペンを起動します。Xアイコンのメニューから起動してもいいですし，シャーペン.Xのアイコンをダブルクリックしても構いません。旧バージョンのシャーペンが起動しないように気をつけてください。

メニューバーのいちばん右の，環境アイコンでマウスの右ボタンを押下し，出てきたメニューから「キー定義の書き出し..」を選びます。するとファイル名入力を促してきます。ここにはおそらく「a:/シャーペン/シャーペン」のようなファイル名が書かれていると思います。これは変更しないで，リターンキーまたは実行ボタンを押します。

シャーペン.Xと同じディレクトリに「シャーペン」という名前のファイルができているはずです。これがキー定義ファイルです。このファイルをシャーペン.Xにドロップして編集します。

どのように編集するかは，シャーペンのマニュアルを見るか，追って説明する各外部コマンドのキー割り当て方法を読んでください。

編集が終わったら，OPT1＋Sを押して保存します。

いちばん右の環境アイコンのメニューから「キー定義の読み込み..」を選びます。ファイル名入力を促しますが，先ほどと同じファイル名が書かれているはずです。そのままリターンキーか実行ボタンを押してください。

キー定義ファイルの編集で間違いがあると，ダイアログが出るはずです。その場合はもう一度正しく編集しましょう。間違いがない場合はなにも起きません。

環境アイコンのメニューから「キー定義と環境の保存..」を選びます。「キー定義ファイル＋.ENV」(かつ，シャーペン.Xの拡張子を.ENVに変えたもの）がすでに書かれているので，リターンキーか実行ボタンを押します。

これでキー定義の変更ができました。念のためOPT1＋Qでシャーペンをすべて終了させ，もう一度シャーペンを起動して試してください。

解説

では，それぞれの解説をします。SX-WINDOW ver.3.1のシャーペンのマニュアルと同じような構成に加え，「コメント」として内部事情などを解説します。

●注釈の挿入（i-f-comment)

書式　i-f-comment［switches］

解説

i-f-commentは，指定コラム位置に指定スタイルで注釈文字を挿入します。タブ文字は8文字とみなします。編集しているファイル名の後尾が“S”“P”のときは“*”，後尾が‘CC’のときは“//”，後尾が‘L’のときは“;”，後尾がそれ以外では“/* */”を挿入します。すでに注釈が存在する場合には，新たに注釈を挿入せずスタイルのみ変更します。

注釈の開始文字列がクォートされている場合にはコメントとして認めません。ただし仕様により，

printf ("/"/*/" is comment./n");

の/*を注釈として処理してしまいます。ちなみにこの例はGNU Emacsでも誤動作します。

「前の文字属性を参照する」にしていると，注釈の終端で改行コードを挿入した場合に新しい行のスタイルが注釈と同じになります。これを回避するには，外部コマンドfreturnを利用します。

Emacs系ではESC;に割り当てられます。

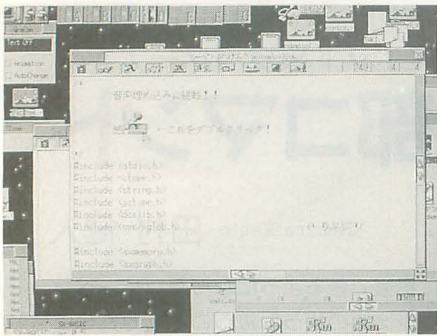
-Cn	コメントの色
-Fn	フォントフェイス
-Kn	フォントID
-Mn	描画モード
-On	挿入コラム
-Sx,y	フォントサイズ

書式例　i-f-comment-*C7-O48*

48コラム目に青色でコメントを挿入します。

コメント

GNU Emacsの‘indent-for-comment’を目指して作りました。とにかくこれは便利で，C言語のソースを入力するときに注釈の入力がとても楽になります。シャーペン以外のエディタで編集するとフォントや色の情報が失われてしまうので，これを使い始めるとシャーペン以外を使うのが嫌になります。

●スタイルの前方参照をする改行コード挿入（freturn)

大技，AD PCMの埋め込み

書式　freturn
解説
freturnは，スタイルを前方（テキスト終端方向）から受け継いで改行コードを挿入します。「前の文字属性を参照する」にチェックがついているときでないと意味がありません。たとえば24ドット文字と16ドット文字が並んでおり，その境目にカーソルがあるときに改行コードを入力したあと，次に入力される文字は24ドット文字となります。fretrunで改行コードを挿入すると，16ドット文字が入力されます。
キー定義ファイルでCTRL＋Mに割り当てただけでは改行キーが変更されません。E0マクロで外部コマンドmapを用いて割り当ててください。'map-K'のところに'，＄0，＄1d，1，＄d'を追加すると改行キーとCTRL＋Mが同じ動作になります。
コメント
やっていることは「#4,M1,'copykind-c'，＄0d,#19,M1,'copykind-v'，＄0d,#13」（右に行って属性を取って，左に戻って属性をセットして改行を挿入）そのものなんですが，これだとカーソルが［EOF］のところにあるときにものすごく変な動作をします。カーソルが［EOF］にあるときに例外処理を行うためだけに外部コマンドになりました。シャーペンのキー定義で条件判断がまったくできないのがとても残念です。

●インクリメンタルサーチ（isearch）
書式　isearch［switch］
解説
isearchは，インクリメンタルサーチを行います。実行中のキー操作は次のようになります。
　CTRL＋S　前方次検索
　CTRL＋R　後方次検索
　CTRL＋G　サーチを始めた位置に戻って終了
　CTRL＋Q　次に押下する文字をそのまま入力
　CTRL＋H，BSサーチ文字列を1字削除
　改行，ESC　その位置で終了
　一般キー　サーチ文字列に追加
テキストの1行目でマッチしても表示が見えないという不具合があります。全域検索はできません。1バイトアルファベットの大文字小文字は区別しません。
Emacs系ではCTRL＋Sに'isearch'，CTRL＋Rに'isearch-b'が割り当てられます。
-B　後方検索
コメント
「インクリメンタルサーチ」について説明します。直訳すると「増加検索」となりますね。これは文字列検索のユーザーインタフェイスです。普通の検索では，
　文字列を入力
　検索
という手順ですが，インクリメンタルサーチでは，
　aを入力
　"a"を検索
　bを入力
　"ab"を検索
　cを入力
　"abc"を検索
となります。文面ではこの便利さは伝わりにくいと思いますから，ぜひ実際に使ってみてください。
GNU Emacsのインクリメンタルサーチとは微妙に動作が異なる部分があると思いますが，私が気にならないのでこれでよしとします。
外部コマンドmapで一般キーにマクロを割り当てると，そのキーの入力がうまくいかないみたいです。謎ですね。

●#記号の挿入（e-c-sharp-sign）
書式　e-c-sharp-sign
解説
e-c-sharp-signは，文字'#'をカーソル位置に挿入します。ただし，行頭よりカーソル位置までスペースとタブしかなかった場合，それらを削除し行頭に#を挿入します。
このコマンドをSHIFT＋3に割り当てるには外部コマンドmapを利用します。E0マクロ中の'map-K'のところに「，＄1，＄04,10,67」を追加し，ユーザー定義コマンド4の4番目を「M1,'e-c-sharp-sign'，＄0D　＊　#」とします。
コメント
C言語のソースを入力するときは「オートインデントモード」にしたほうが楽です。しかしこの状態では「この行に#ifdef……と書きたいのだがカーソルはすでに行頭にない」という状況が出てきます。e-c-sharp-signを使えばオートインデントの直後でも#を行頭に入力してくれるのです。

●タブ，スペースの最適化（optab）
書式　optab
解説
optabは行頭よりカーソル位置までのタブ，スペースをもっとも効率のよい状態に置き換えます。行頭からカーソル位置の間にタブ，スペース以外の文字がある場合には機能しません。タブはスペース8文字ぶんとみなします。
例:>がタブ，_がスペースだとします
　>____>strlen（_buf_）；
この状態でoptabを実行すると，
　>>strlen（_buf_）；
となる。
キー定義ファイルで，「M1,'tab'，＄0D＊^I」となっている部分を「M1,'tab'，＄0D,M1,'optab'，＄0D＊^I」と書き換えると，CTRL＋I/TABでタブを入力したときは必ず機能するようになります。
コメント
行頭からカーソルの間に，スペースが1個だけある状態を想像してください。そこでタブキーを押したらどうなるでしょう？　気分的に「いや～ん」な状態になるのがわかりますね。これを解消します。

●OPMへの書き出し（wopm）
書式　wopm
解説
wopmは，セレクト範囲があればセレクト範囲を，なければ全体を'opm'に書き出します。

●右括弧に対応する左括弧の表示（rpar）
書式　rpar［switch］
解説
rparは，カーソル位置の左にある右括弧に対応する左括弧を指摘します。具体的には，対応する左括弧を0.5秒間反転します。')'が入力されたときに常にこのコマンドが実行されるようにするには外部コマンドmapを利用します。E0マクロ中の'map-K'のところに「，＄1，＄0A,10,73」を追加し，ユーザー定義コマンド4の10番目を「')'，M1,'rpar'，＄0D　＊)」とします。
「（＊p　＝＝　'/'）」の最後の右括弧に対応する，最初にある左括弧を認識できないという不具合があります。
　-n　反転する時間をn［x1/100sec］にする。初期値は50
書式例　rpar 40
カーソル位置の左にある右括弧に対応する左括弧を0.4秒間反転します。

コメント

複雑な式を書いたときに括弧の対応があっているかどうかがわかりやすくなります。

特にシャーペンでLISPのソースを書くことのあるごく一部の人は重宝することでしょう。

● '}'記号のインデント（e-c-brace）

書式　e-c-brace　[switch]

解説

e-c-braceは，rparと同様カーソル位置の左にある'}'に対応する'{'を指摘します。さらに，もし'}'から行頭までタブとスペースしかない場合，対応する'{'がある行と同じインデントに変更します。

```
    if  ( flag )   {
        func1 () ;
        func2 () ;
              } ▊
```

この状態でe-c-braceを実行すると，

```
    if  ( flag )   {
        func1 () ;
        func2 () ;
    } ▊
```

となります。また，

```
    if  ( flag )   {
        func1 () ;
        func2 () ;
} ▊
```

この状態でe-c-braceを実行しても同じ結果となります。

'}'が入力されたときに常にこのコマンドが実行されるようにするには外部コマンドmapを利用します。E0マクロ中の'map -K'のところに「,＄1,＄29,10,125」を追加し，ユーザー定義コマンド4の最後から3番目を「'}',M1,'e-c-brace',＄0D ＊]」とします。

n　反転する時間をn [x1/100sec] にする。初期値は50

●データ付きイメージの作成（xpaste）

書式　xpaste

xpaste 〈type〉〈ファイル名〉

解説

xpasteはセレクト範囲のイメージに任意のデータを張り込みます。引数なしで実行した場合，スクラップの最初にあるデータ（クリップボードのリスト表示でいちばん上のデータ）をそのまま張り込みます。

xclickでPCMとして鳴らすときには，文書中にあるイメージを選択し，タイプ'xpcm'でAD PCMデータをペーストします。

〈type〉はアルファベットの大文字小文字を区別します。〈type〉についてはxclickの解説を参照してください。

書式例　xpaste xpcm a:/adpcm/makeup.pcm

セレクトされたイメージにmakeup.pcmをxclickで鳴らせる形式で張り込みます。

●ダブルクリックの拡張（xclick）

書成　xclick

解説

xclickは，xpasteによって作られた文書中のデータ付きイメージをダブルクリックしたときの動作を拡張します。S0マクロで実行します。S0マクロは標準のキー定義ファイルでは「…M1,'kei',＄0D」となっていると思います。そこを「…M1,'kei',＄0D,M1,'xclick',＄0D」のように変更します。

xpasteコマンドによって作られたxpcmタイプのデータ付きイメージをダブルクリックすると，PCMを鳴らします。ただし，そのイメージをダブルクリックしたとき，セレクト範囲中に「絵」という文字があるとなにも起きないことがあります。PCMDRV.SYSもしくはそれに相当するものが組み込まれている必要があります。

環境変数tempを参照することがあります。

xpasteで埋め込むタイプと動作の関係を以下に示します。

・xpcm

'pcm'にコピー

・xx_xr_

データをX型・R型の実行ファイルとして起動

・xpen xdoc xtxt xref xman

データを読み込んだシャーペンを起動する

・xzms xopm xzmd xmdx xmdn xmdf xrcp xr36 xmcp xmdz xmdi xzdf xmid xmff xstd xkdd

データをsxzc.rで演奏する

コメント

xpaste&xclickでシャーペンの可能性は一気に拡がります。キー定義ファイルを編集しないで簡単に体験するには，

0)　PCMDRV.SYSに相当するものを組み込んでSX-WINDOWを起動

1)　付録ディスクに入っているxclick.cをシャーペンに読み込ませる

2)　OPT1＋1を押してから，xclick.exをそこにドロップしてリターンキーを押す

3)　PCMのアイコンが見えるはずなので，それをダブルクリック

とします。どうです？　音が鳴りましたか？

この外部コマンドは「イメージに網掛けをしてもなにも変化がない」というテキストマネージャの仕様を利用したものです。先月号に書いた「常にイベントを乗っ取る」ということも，なんとか実現できました。本当にこれで不都合が起きないのかどうかわかりませんが，私の環境や編集部では特に問題は起きていません。

もともとPCMを鳴らすために作った外部コマンドですが，調子に乗ってシャーペンテキストと曲データにも対応させてみました。「ここまでやるなら任意の拡張子に対応させて，それぞれのアイコンの設定を見るようにしようか」とも思ったんですが，環境に依存した動作をするのは好ましくないと考えて決まったソフトを起動するようにしました。

xclickに対応したデータを詳しく解説します。ここではあるイメージ(PAT4でもドローでもなんでもよい）にテキストファイルa:/text/第1章.penを張り込む手順を述べます。

0)　EXCOMディレクトリにxclick.ex xpaste.exがあることを確認

1)　シャーペン上で，目的のイメージを選択

2)　OPT1＋1を押して,「xpaste xpen　」と入力する

3)　a:/text/第1章.penをドロップする

4)　「xpaste xpen a:/text/第1章.pen」となっていることを確認し，リターンキーを押す

以上で完了です。張り込まれたデータは変更ができないので，内容を変えたいときはもう一度この手順を繰り返す必要があります。張り込まれるのはファイル名ではなく，データそのものなのです。

総評

早い時期にi-f-comment.exを作って色つき注釈の虜となってしまったため，それ以後ソースを書くのはシャーペンオンリーになってしまいました。

それからは，

1)　外部コマンドのソースを入力する

2)　ソースの入力が不便だと思う

3)　便利にするために外部コマンドを作る

の好循環です。

xpaste&xclickを自己評価すると「満点」です。我ながらよくやったと思っているのですが，皆さんの評価はどうでしょうか？

今回収録した外部コマンドのソースは，どんどん再利用して結構です。これを参考に皆さんも怪しい外部コマンドを作ってみてください。

IO-735, MJ700V2C対応

カラーハードコピーツール

Taki Yasushi 瀧 康史

美しいグラフィック画像と高性能なプリンタ
せっかくの素材ですからできるだけ有意義に使えるようにしてみましょう
このようなツールがあってこそカラープリンタを活かすことができます

カラー印刷

初めてカラー印刷が，いいなと思ったのは，知人のエプソンのインクジェットプリンタを見たとき，もはや名前もスペックも覚えていない。ホストコンピュータはデジタル8色のPC-8801。mkIIもSRもつかない，ただの88だ。

デジタル8色ということは，RGBそれぞれ1ビットずつなので，プリンタにCMY変換したときにそのまま印刷できる。減色処理もなにもいらない。解像度はおそらく160dpiぐらいなんじゃないだろうか？ ひょっとしたらもっと悪かったかもしれない。

しかし，それでも私を興奮させるなにかがあった。やがて自分でPC-PR101TL(熱転写プリンタ)を購入してアナログ4096色のデータをグラフィックツールで印刷して幻滅する。ああ，こんなもんなのかもしれないなと。

数年後，Oh!X編集部に出入りするようになる。初めて出入りするようになったのは，ちょうど3年ぐらい前の印刷特集のまっただ中。当時の編集部はIO-735Xを利用して，PICファイルをポスターサイズでガンガンと印刷してはベタベタとアニメショップのように壁に貼っていた。正直，ちょっと怖いものがあったが，カラーインクジェットプリンタに対しての憧れを，多少取り戻すことができた。数カ月後，IO-735Xを購入することになる。

印刷特集のとき使用された浜崎氏と中野氏の合作BASICプログラムをもらい，Cにしてオプティマイズし直す。ファイル関係の取り扱い，メモリ管理をC風に起こし直す。なぜか，年2回のイベントの前に印刷を頼まれるため，そのたびにバージョンアップを繰り返していたが，ともかく，IO-735Xの印刷プログラムとしては，それなりに完成しているものができあがった。

ハッキリいって，いまとなってはIO-735Xは解像度も180dpiと粗く，見た目の数字だけを考えてプリンタ選びをしている人には魅力的ではないかもしれない。

今回もみじ狩りPRO-68Kに収録された印刷プログラムを使ってみれば，再びIO-735Xを見直すだろう。このプログラムは，IO-730でも動作するはずである。現在，IO-730なら(印字性能はほとんど同程度で)中古が1万円強で買える。ポスターサイズでカラー印刷できるのは，やはりポイントが高い。置き場所にさえ余裕があるならば，購入してみることをおすすめする。インクはまだ取り扱われているのだから。

さて，先月の約束どおり，まともなものができるまでのつなぎではあるけれど，とにかくMJ700V2Cの印刷プログラムが完成した。基礎となる部分はIO-735Xの印刷プログラムと同じで，まだMJ700V2Cの機能を半分も使い切っていないと感じているが，とりあえず現時点で完成しているバージョンをお届けしたいと思う。

印刷プログラム

浜崎氏がアルゴリズムなどを説明してから随分と月日がたってしまったので，改めてプログラムの原理の説明をしよう。

IO-735X，MJ700V2Cいずれの印刷プログラムとも，スケルトンは同じである。元のプログラムは，浜崎氏の作成したBASICプログラムで，これにBtoCをかけ，自分にわかりやすいように手直しやオプティマイズを加えた。HC735light.C，HC735.C，HCESCV2X.Cという順序でプログラムを開発していったので，新しいほど性能がよい。

それではHC735light.Cから説明をしよう。

まずHC735light.Cは(H)ard (C)opy IO-(735)X (Light)Memory versionの略。文字どおり，IO-735Xにて小メモリ環境で印刷をするもの。完全に確認しているわけではないがメモリフリーが1Mバイトぐらいあれば印刷できると思われる。

印刷時間はX68030において40分から50分ぐらい。10MHzのX68000ではおそらく2時間ぐらいかかると思われる。プリンタドライバのバッファ指定を大き目にとっておけばかなり改善されるのでできるだけ，

DEVICE=PRNDRV1.SYS #/B512

のような指定を加えておいてほしい。

3つの印刷プログラムは，どれも減色処理に粂野式拡散法を利用している。粂野式はアセンブラには相性のよいアルゴリズムだが，Cには特に当たり触りのないアルゴリズムである。アルゴリズムに関しては後ろを参考にしていただきたい。「速く印刷したければよいマシンを買おう！」という発想の下に作られているので「とても」遅い。印刷クオリティを多少落とすなり，アセンブラ化してしまえば，高速な印字方法がないわけではないが，もはやこのプログラムに手をかける気力もなくなってしまったので，このままでいる。

このHC735light.Cでもそれなりの印刷を楽しめるが，これをIO-735Xのポスターサイズで印刷すると，1ドットが結果的に1.5mm×1.5mmぐらいになり，少々ドットの粗さが気になるときがある。プログラムを最小限の改良で量子を細かくするように工夫したのが，HC735.Cになる。

印刷時間はHC735light.Cより多少速く，画像が量子的に綺麗になったのが，HC735.Cである。

まず，一度512×512の3万色の画像を，直線補間法にて，1024×1024の3万色に拡大する。なにも考えずに2倍に拡大するので，メインメモリ中に仮想VRAMとして2Mバイト必要になる。したがってメモリ4MバイトのマシンでないとHC735.C，HCESCV2X.Cは動かないだろう。仮想VRAMをメインメモリに持つ副作用として印刷中

にVRAMの内容を変えてもよくなった。また，印刷中，基本的な部分をユーザーモードでアクセスできるようになったので(VRAMのアクセスはスーパーバイザーモード)，SX-WINDOWのシャーペンコンソールで実画面背景をバックグラウンドで印刷できるようになった。

余談ではあるが，シャーペンのコンソールはユーザーモードでないとタスクの切り換えをしない。したがって，大半がユーザーモードで動くというのはメリットになるわけだ。

余談はさておき，MJ700V2C用の印刷プログラムである。基本的にHC735.Cをこちょこちょといじったら動いてしまったので，それなりの機能しかない。ただ，ESC/P V2の機能で，MJ700V2C用に追加された機能の効率的な使い方がイマイチわからず，720×360dpiになっている。インクの滲み率もまだまだ調査段階で，エプソンのWINDOWS版プリンタドライバに比べるとはるかに貧弱な印刷しかできない。PICファイル1枚分程度の解像度の印刷としては，それでも最強クラスか？　まぁこれらの不都合な点は後ろの項で書くことにしよう。

プログラムオプション

どのプログラムも基本的にはデフォルトで，紙の最大サイズを出そうする。なにもオプションを指定しないと，なにもいわず戻ってくるが，-?などをオプションにすると，ヘルプを表示する。

基本的には，下記のような表記でコマンドラインから打ち込む。

A>HCxxxxx　[OPTION……]
[FILENAME……]　[OPTION……]
[FILENAME……]

OPTIONは，次の3つが共通してある。

-Sn サイズを指定する。
　n=0 ノーマルサイズ（小さい）
　n=1 ワイドサイズ
　n=2 正方形ドット
　n=3 縦印刷モード
-Mn 色分解の指定
　n=0 3色分解
　n=1 4色分解
　n=2 4色分解＋色相変換
-N　画面を初期化せずに印刷

なお，ファイルネームは，PICファイルを指定する。内部でAPICを呼んでいる。APICを持っていないなら，適当に画像を読んでから，「本来ファイル名を書くべきところに」-Nオプションを入れること。基本的にシェルの許す限りいくつも書いてよい。実行ファイルはHUPAIRに対応しているので，fishなどのSHELLから実行すれば，メタキャラクタを利用して連続して多量に印刷ができる。

なお，オプション指定にスラッシュ［/］は許していない。これは，ファイル名と間違えるためである。基本的にオプションはそれ以降に列挙した，つまり右にあるファイルのみしか影響しない。

各オプションについて説明しよう。

まず，サイズである。

サイズは0，1，2，3と適当に入れてあるが，それほど根拠のあるサイズではない。秘かに，HCESCV2Xでは好きな大きさに指定できるようにしてある。隠し機能なのでエラーチェックなどしていないから，自分でソースなどを見て解析してほしい。なお，同じくHCESCV2Xのみ，画面の指定の範囲のみを印刷する機能をつけたような気もするが，気のせいかもしれない。もし見つけても，バグがあるかもしれないので使わないように。

どのプログラムにも共通しているサイズは最大サイズの1，2，3である。0は実験時に印字しやすいサイズにしてある。これは私ぐらいしか使わないサイズだろう。

おそらくイラストなどをカラー印刷するときには，紙いっぱいに印刷したいだろう。というわけで，ファイル名だけ指定したときは，HC735，HC735lightはポスターサイズ(15インチの紙幅を画面の縦のサイズで印刷する)で，HCESCV2XはA4の紙にギリギリ入る大きさで印刷をする。

サイズ2は横幅ギリギリサイズの正方形。

サイズ1は横幅を画面の横幅にしたサイズ。HCESCV2Xでは下が結局余ってしまうので意味はないかな。

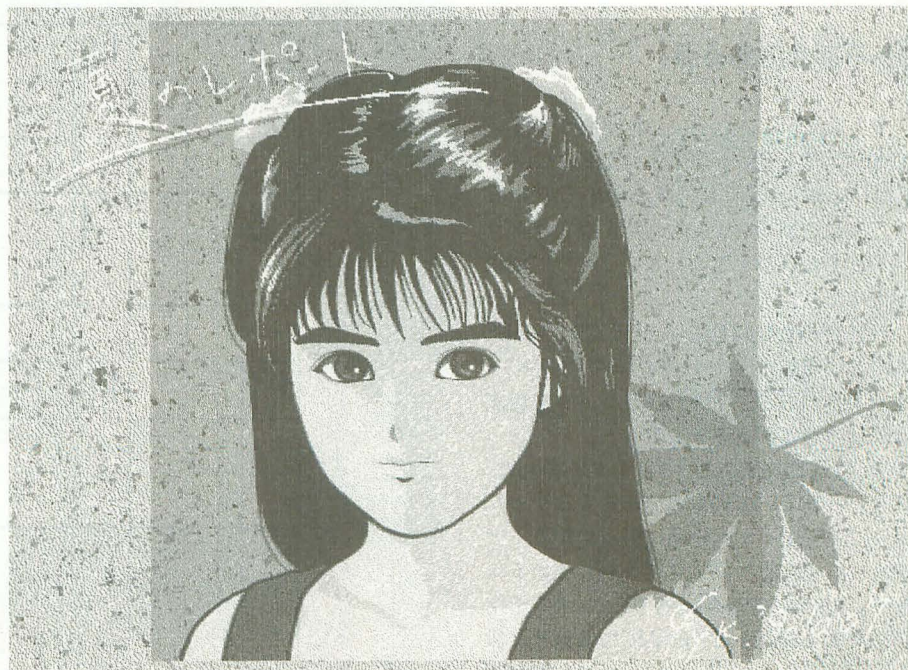
次に変換モード(Mオプション)についての話をしよう。原理は後ろの項で解説するとして，ここでは結果を述べることにする。

まず0は3色分解。RGBをCMYにして，3色でそのまま印刷する。ハッキリいって，黒がしっかり出さえすれば綺麗。実際問題として，3色をうまく混ぜても黒がうまくできないし，エッジが滲みがちになる。

最新のMJ700V2Cのほうはさほどでもないが，IO-735Xはなにぶん古いプリンタなため，3色分解モードで黒が多くなると，インクがダラダラと流れてくる。流れ出さないにしても，滲んだり，黒に偏りができやすい。

そして1がこれらを解決した4色分解。HC735，HC735lightとHCESCV2Cの4色分解のアルゴリズムは多少違う。

モード2が色相変換を踏まえた4色分解で，これがあるのは，HC735，HC735lightのみ。MJ700V2Cはこれがないぶん，個人的に強化させた4色分解を行っている。

-Nオプションはほかとは少し性格の違うオプションで，画面を初期化せずにファイルロードもせず，いまVRAMにある画面を印刷する。主に，グラフィックツールなどから子プロセスで呼び出せるようにとつけてあるオプションだったが，HCESCV2C，HC735ではシャーペンのコンソールモ

PIC画像を原寸で出力

ードでも使えるように多少プログラムに，小細工をしている。

基本的に-Nオプションは，ファイル名の代わりに指定するので，場所的に-Nはファイル名を列挙するべき場所に置かなくてはいけない。

HCESCV2Xはさらに，-Cオプション，-Bオプションがある。このプログラムは印刷に時間がかかるので，普通は印刷したらモニタの電源を切っておくものだが，HCESCV2Xの場合は消し忘れたときのために，デフォルトでスクリーンセイバーが働くようになっている。HCESCV2Xを起動したとき，RGB型直線補間を行った直後にいきなり画面のコントラストが落ちるが，これは意図的にやっているのだ。一時的に画面の様子を見たい場合，コード入力キーを押していただければ，押している間はコントラストがSRAMの設定値までアップする。

このコントラストダウンをしないようにさせるのが-Cオプションである。SXのコンソールで背景を印刷するときは，勝手にコントラストを下げられては困るだろうから，必ず-Cオプションを指定するように。

なお，コントラストダウン中にブレイクキーやインタラプトスイッチで作業を停止した場合，コマンドシェル上ならSCREENとタッチタイプすれば，コントラストはSRAM設定値まで上がる。

-Bオプションは色合の設定。設定をいじると，インク濃度を変えられる。100を中心にしていたので，101以上で黒っぽくなり，99以下で白っぽくなる。

不都合のある点

どちらもPICファイルしか印刷できない。MAGもPIもPICもJPEGもTIFもGIFもBMPもQ0もPICTも綺麗に印刷したいなら，もはや別の，まったく新たなプログラムを作らねばならない。というわけで現在開発中だ。

16色データに関しては，HC735とHCESCV2Xは内部に1024×1024，3万色のバッファを持つので，そこに16色データを展開すればうまく印刷できるだろう。

実はHCESCV2Xだが720dpi×720dpiではない。うまい印刷方法がわからなかったからだ。ちょっと頭の隅に引っかかることがあるので，「新たにプログラムしているほうで」720dpi×720dpiをサポートする予定。色の染まり具合もMJ700V2Cにあわせた設定がまだ発見できていない。どうも黄色が薄い。HCESCV2Xは，最初に#defineでCMYBの閾値，インク濃度が設定できるので，適当にいじってコンパイルし直してほしい。閾値は低いほうが濃くなり，濃度は高いほうが濃くなる。なお，プログラムはすべてLIBCでコンパイルできる。

よりよい印刷を求めて

というわけで，よりよい印刷を求めるためのアルゴリズムの解説をしたいと思う。現在制作中のプログラムは，以下のアルゴリズムによってプログラムをしていると考えてほしい。

ひと口に印刷といっても，実際には以下のような処理を踏まえて印刷を行わなくてはいけない。

1) 元絵からプリンタサイズへ解像度変換
2) 減色処理
3) 4色分解（黒抽出）
4) 場合によっては色相変換
5) 実際にプリンタへ出力

今回発表したプログラムは，メモリが少なくても動くぶん，元絵のサイズが固定されている。これらをフレックスにした印刷プログラムを作る場合，私の頭ではどうしても，「効率的に速くメモリを少なく」するアルゴリズムを生み出すことはできない。悪いが，大発見がない限り，いま作成している印刷プログラムはメモリをバカスカ喰い，テンポラリにバカスカ落とすと考えてもらいたい（もっともMS-WINDOWS版でも同じようだけど）。

それでは処理に関することを，順に説明していこう。

解像度変換

プリンタのサイズはプリンタによって違う。できればプリンタ依存のプログラムにしたくないし，最低でも最初の部分を少し変えれば動くようにしたい。そこで，出力先の解像度も任意でなくてはいけない。

参考までに，IO-735Xはポスターサイズで2500×3350ぐらい。MJ700V2CはA4サイズで6000×8000ぐらい。どうにも，解像度でこんなにも違うもんである。

さて，まず印刷するものが決まれば，出力されるプリンタにあわせて解像度変換，いわゆる拡大縮小をしなくてはいけない。この解像度変換によって随分出力の雰囲気が変わるので，このプロセスをおろそかにはできない。解像度変換の部分はライブラリ形式に作れば，いくらでも好きな方法が選べるようになるだろう。現在使用しているものは次の2つの方法である。

1) ゼロ補間法
2) 直線補間法

1)のゼロ補間はいちばん一般的な方法である。まず図1を見てほしい。なかに書かれている数字はドットの輝度と考えてほしい。このような3×3の画像を3倍に，9×9の画像に拡大したいとき，どうすればよ

図1　解像度変換前のデータ例

20	10	15
6	4	30
11	10	25

いのだろうか？

ゼロ補間はもっとも簡単なもので，図２のようになる。見てのとおり，足りない部分を左上のドットと同じもので埋めているだけである。これだと，計算量は非常に軽くなるし，逐次的に計算ができる。今回の３つの印刷プログラムでは，逐次的にプリンタの解像度へゼロ補間を用いて解像度変換している。

このゼロ補間の欠点は画質が粗くなることだろう。当然，１ドットが３×３ドットになってしまうぶん，四角くなってしまい，角が目立つ絵になってしまう。

これを解消するための方法として，2)の直線補間法がある。これは図３を見てほしい。間の小さな数は暗算で求めたので，間違っているかもしれない。見てわかるとおり，輝度，色相などを滑らかにつなぐ処理をしている。

これを行うと，１ドットが四角くはならないため，元画像によってはかなりスムーズに綺麗に再現できる。しかしながら，この方法にも相性があって，写真などのもともと１ドットをあまり意識しない画像はかなり自然になるものの，１ドットが明確にわかる，ジャギーなどが残っているイラストでは，ぼかしたような感じになってしまう。一般に元画像の画素が失われているぼかしよりはもちろんマトモではあるが，それでもエッジがぼけた感じはする。ジャギーを綺麗に消してあるイラストであった場合は，補間によるボケは気にはならないだろう。

図３中，？のところがあるが，これは直線補間法の宿命で，先のドットがない限りドットが予測できないという欠点がある。今回，HC735.CとHCESCV2.Cは，一度512×512の画像から，1024×1024の画像へ直線補間法により画素情報を増やしているが，右と下のエッジはゼロ補間でごまかしている。２つの印刷プログラムは，こうしてできた1024×1024の画像に対して，プリンタの解像度へゼロ補間を用いて変換しながら印刷しているわけだ。

減色処理

それなりの本を探せば減色処理はいろいろあるのだが，メジャーなものとして，誤差拡散，ディザ変換などがある。Oh!Xでは桒野式拡散という伝統的な変換があって，私自身，この変換は割合気にいっている。

桒野式拡散は，私にとっては大先輩である桒野氏が試行錯誤によって編み出した（と思われる）方式である。

たとえば，ある解像度のグラフィックを単純に減色するときにはどうするだろうか。

ここで，図４のような輝度を持つ画素が分布されている画像があったとしよう。ちなみにでっち上げの情報なので，どこかの画像から取ってきたなどといった深い意味はない。

これを単純閾値法により減色処理をしてみる。無謀にも０と１に分けよう。こうなった場合，閾値を決定しなくてはならない。半分ということで，閾値Ｔは15ということにしよう。15以上が１，14以下が０とすると，図５のようになる。

当然，このように大胆に切られてしまう。失われる画素は多く，かわいそうな13さんは，見事０になってしまった。もしも輝度を上げるために閾値Ｔを13にしたら，13は救われるというわけだ。

そこで，敗者復活作戦というわけで，無作為に天から雲の糸を下ろす。といってもまったく無作為ではなく，ある一定の規則に従って下ろすのが減色法というやつだ。

まず，桒野式だが，図６のような式によって算出する。たとえば，左上の31は右に31/2＝15，下に31/8＋(31 mod 8)＝3＋7＝10，右下に31/8＝3と輝度を分ける。これを累算していくと実際には，31の右のドット

図２　ゼロ補間法

20	20	20	10	10	10	15	15	15
20	20	20	10	10	10	15	15	15
20	20	20	10	10	10	15	15	15
6	6	6	4	4	4	30	30	30
6	6	6	4	4	4	30	30	30
6	6	6	4	4	4	30	30	30
11	11	11	10	10	10	25	25	25
11	11	11	10	10	10	25	25	25
11	11	11	10	10	10	25	25	25

図３　直線補間法

20	16	13	10	12	14	15	?	?
15	13	11	8	12	16	20	?	?
11	9	8	6	13	20	25	?	?
6	5	5	4	13	22	30	?	?
8	7	7	6	14	21	29	?	?
10	9	9	8	14	21	27	?	?
11	11	10	10	15	20	25	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?

などは15+8=23となり，単純閾値法では捨てられたこのドットは1となり救われる。

データは累積的に計算されるため，左から(2,2)の位置にいる25というドットは，

25+31/8+23/8+(23 mod 8)+14/4=44

となる。

同様に計算していきたいところだが，見てすぐわかるとおり，結構計算がややこしい。表は手計算なのでこれ以上は勘弁してほしい。

桒野式の長所は，アセンブラにしやすいアルゴリズムであることだ。n/8,n/4,n/2は，いずれもアセンブラでは数学的に右に3シフト，2シフト，1シフトであるし，n/8+(n mod 8)は8ビット的に上バイトと下バイトを足してやればよい。

というわけで，みっちりとアセンブラ寄りで，結構簡単なアルゴリズムの割には綺麗な印刷が楽しめる。アセンブラでプログラムを組めば，速度アップは間違いないだろう。

カラーのときはどうするの？　といわれそうだが，カラーでも考えることは同じである。具体的には，最初にRGBをインバースしてCMY化する。そして，それぞれの色で処理を行えばそれでよい。これで3色分解（8色）になる。

4色分解（黒抽出）

基本的にたいていの色は3色の合成で近似表現が可能である。しかし，たいていのカラープリンタには4色目に黒がある。

黒というのは理想的CMYをすべてを混ぜた色だ。ただCMYの3つが，完全にほかのスペクトルを吸収しないかぎり完全な黒にはならない。そこで，カラー印字を刷る際，黒インクを効果的に使う（もっとも，一般にプリンタに黒があるのは黒がいちばん使われるからだろうが）。

図8を見てほしい。図の共通部分が黒としての濃度である。この黒の部分だけを抽出すると，4色に分解できることになる。

基本理論はこれで終わりだが，実際はもう少し小細工が必要になる。

まず，今回扱うプリンタに中間濃度というのはないということ。それゆえ，薄い黒といっても灰色が打たれるわけではなく，黒を少しだけ拡散し，黒が斑に点在する状態になる。淡い部分はかなりまばらに点在することになるが，イエローやシアンと違って，点在しているとくっきりと目立つのが黒である。

ハッキリいうと，MJ700V2Cのような滲みにくく，色乗りのよいプリンタならば，ある点までは3色分解がよい。4色分解すると多少遠目でも黒の点在が見えてしまうが，3色ではうまくぼけるからだ。

そこで，黒の抽出は図8のように安易に抽出するのではなく，取り出し方に一定の傾斜を加える。

まず多めに黒を抜き出してほしいのはどんなときだろう？　当然，真っ黒のとき。これは当たり前。そうでなく傾向的に表すとしたら？

1)　全体の色が黒めのときは，黒を多く抽出してほしい。

2)　灰色の系統の色に近いほど，黒を多く抽出してほしい。

私が適当に考えるかぎり，この2つが持ち上がる。

1)の逆例，黒めの色じゃないとき，たとえば淡いシアンだったとしよう。このときのカラーコードが，CMY順に256階調で40, 10, 10となる。これをCMYB4色に分解すると，30,0,0,10となり，普通の4色分解では，黒が10抽出されてしまう。

淡いシアンを拡散するとき，できるならば，あまり黒いポツポツは出てほしくない。そういうわけで，黒の抽出に傾きを与えてできるだけ黒を抽出しないでほしいと思う。

次に2)の例。

たとえばCMY順に10, 10, 10だったとしよう。色は灰色だが，1)の傾斜を踏まえてしまうと，悲しいことに，CMYBにしても，Bにわずかしか拡散されない。むしろ，

図4　画像の一例

31	8	3	16	
5	25	13	4	
21	12	23	6	
10	2	7	20	

図5　単純閾値法による減色処理

1	0	0	1	
0	1	0	0	
1	0	1	0	
0	0	0	1	

図6　桒野式拡散法の考え方

図7　桒野式拡散法による拡散分布

CMYには少なく，Bに多く拡散してほしい。

結局上の2者は，どちらも適用されなくてはいけない。CMY，個々の濃度の最低値をmin，最大値と最小値の差をdifとしよう。これを式にすると，1は，

$$B1=min\times\frac{min}{255}$$

になる。255は256階調時の最大値。なぜmin/255が黒に近いことを表す式かは，説明する必要もないだろう。

そして2)は，

$$B2=min\times\frac{255-dif}{255}$$

となる。difは最大値と最小値を指すのでこれが0のときは，完全な灰色を示す。この1)と2)のケースがどちらも適用されるためには，

Black＝B1＋B2

になる必要がある。

Black＞minではいけないので，minを超えたときはminに補正する。場合によっては，

Black＝(B1＋B2)/2

にしたほうがよいかもしれないが，4，5回の実験結果からは，“/2”はないほうがよいのではないかと思っている。

別に，著名人が作ったわけでもなんでもない単なるでっち上げ理論だが，素直な4色分解に比べて，3色分解と4色分解の長所を兼ねあわせたような変換をしてくれる。

個人的には割と満足しているのだが……

色相変換

色相変換はインクの滲み率と，RGBの発色が違うことから行う変換である。

どこかの本でちらっと読み，メモ帳にメモしたはずがなくしてしまったという状態だが(つまりうろ覚え状態)，図9を見てほしい。

人間の可視範囲の色は図の曲線で書かれたような範囲になるそうな。RGBモニタというのは，その中の大きめの三角形で，それぞれの頂点がそのRGBモニタの発色限界を示している。この三角形の外はそのモニタでは，発色できない色。発色のよいモニタというのは，この頂点が広がっているってことになるな。大昔のモニタは赤が綺麗だとか，そういう風に載っていたが，最近はドットの量子ばかりカタログに記載されていて，発色についてはなにも載っていないのが残念かもしれない。

さて，RGBモニタはかなりよいところまで発色できているそうだけど，印刷物に関してはまだまだ三角形が狭いそうである。これは，Oh!Xのカラー写真などについても同じ。RGBよりも発色はよくない(編注：基本的に写真のほうがCRTよりも色帯域は広い)。

プリンタの印刷など720dpiになっても，まだまだ。狭い三角形だろう。ひょっとしたら，図の小さな三角形のように少し歪んでいるかもしれない。

そういうわけで，必ずしもRGBの反転がちょうどよい色というわけではなくなってくる。特に青なんかはMJ700をもってしても，なかなか綺麗に発色しない。グラデーションを印刷すると，この青はこの位置にきてほしいなぁという部分があるので，きっと青に関していえば，大きく歪んでいるのだろう。

そこで，ある程度のテーブルをもって，色相の変化を予想する方法がある。これは数値的に計算できるものではなく，ほとんどノウハウによるものだと考えてよい。

HC735.C，HC735light.Cは，中野氏が昔，努力の挙句，かなりよいところまで，近づけた，ただこちらのほうは黒抽出が単純なものであるため，どうも色の混ざりがよくない。

これは今後，どのように解決するかわからないし，手がかりもつかめていない。イメージスキャナで一度印刷したものを取り込み，うんたらと考えたが，中野氏が当時それに挑戦してみたところ，うまくいかなかったそうだ。

最終的には地道な努力がものをいう分野ではあるかもしれない。

おわりに

いろいろやった挙句，MJ700V2Cの印刷はプログラムが完成したらもうやらなくなるかもしれないと思った。というのも，A4サイズで印刷してもつまらないのだ。やっぱりポスターサイズがいいよなぁ。

結局，理論先行，プログラム後回しの原稿でした。グラフィックを印刷するのではなく，グラデーションを印刷するプログラムなどをつくって実験を重ねてみた結果，黒抽出アルゴリズムは作り出しました。

私は画像専門の人間ではないため，イマイチわからないところもあるし，困ったことに数学がとっても嫌い。

「そういうのはですねぇ。根性ですよ」

と多分，本人は覚えてないが，昔石上さんにそういわれたことがある。

いつ，いま作成中のプログラムをお届けできるかわからないが，それまでは，つなぎで我慢していただきたい。

プリンタへの依存をできるかぎりなくしたプログラムにするので，ほかのプリンタの人も期待してほしい。ターゲットプログラムは，

メモリとCPUパワーがなくても印刷できるもの

メモリとCPUパワーを考えずに印刷の綺麗さを追求したもの

の2つ。どこまでいけるかは私の根性次第だったりする。

図8　黒抽出の実際

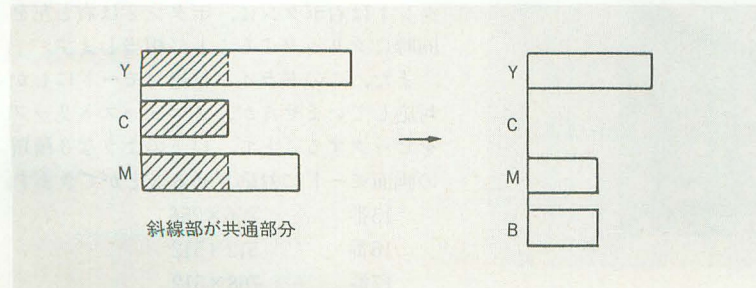
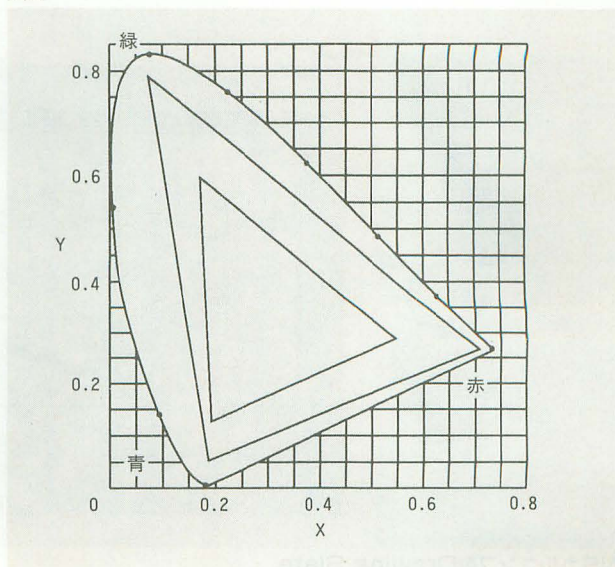



図9

マウスの代わりにペン操作

タブレットマウスドライバ

Kikuchi Isao 菊地 功

これまで専用プログラムでしか作動しなかったタブレット
これをマウスの代わりとして使用するためのドライバです
お絵描き以外にもいろいろ使えます

ちょっと前の話になりますが，NSカルコンプから軽量高機能なDrawing Slateというタブレットが発売されていました。MATIERなどに対応しており，すでに購入されている方も多いと思います。

しかし，タブレットの接続やコマンドなどには特に規格はなく，残念ながらその製品に対応したソフトでしか使用することができません。それ以外のソフトでは使わないというのであれば構わないのですが，フリーウェアや，ちょっとしたマウスを使うツールでタブレットを使いたいという場合があるかもしれません。

そこでタブレットマウスドライバです。

そもそも当初はEX-WINDOWをタブレットに対応させようというところからきたのですが，外部ファイルなどの関係もあり，タブレットでマウスをエミュレートするドライバを作ることになったわけです。

機能

このドライバは，前述のDrawing Slateおよびその前の製品であるDrawing Board-IIとDrawing Padに対応しています。が，手元に旧機種がなかったためにDrawing Board-IIとDrawing Padについては動作確認できませんでした。

というわけで，ひょっとしたらその2つに関しては正常動作しないかもしれませんので，ご了承ください。通信条件はタブレットに付属のデモと同様ですので，そちらのドキュメントを参照してください。

タブレット自体は筆圧や傾きなどを検知できるものなのですが，マウスをエミュレートさせるということで，今回は座標とボタンのみに対応しています。

具体的な動作としては，IOCSのマウス割り込みをトラップして，タブレットからのデータをIOCSのワークに格納しているのですが，マウスからやってくるデータも考慮していますので，特に切り替えのスイッチなどなしでタブレットとマウスを同時に使用できます。

ただし，IOCSレベルのエミュレートですので，IOCSコールを使用しないソフトでは動作しません。手近なもので試したところ，VSは動作しましたが，WPやSX-WINDOWでは動作しませんでした。WPで動かないのはちょっと意外だったのですが，SX-WINDOWで動作しないのは残念です。また，ソフトウェアキーボードには対応していません。

使用法

コマンドラインから，単に，

CalTab

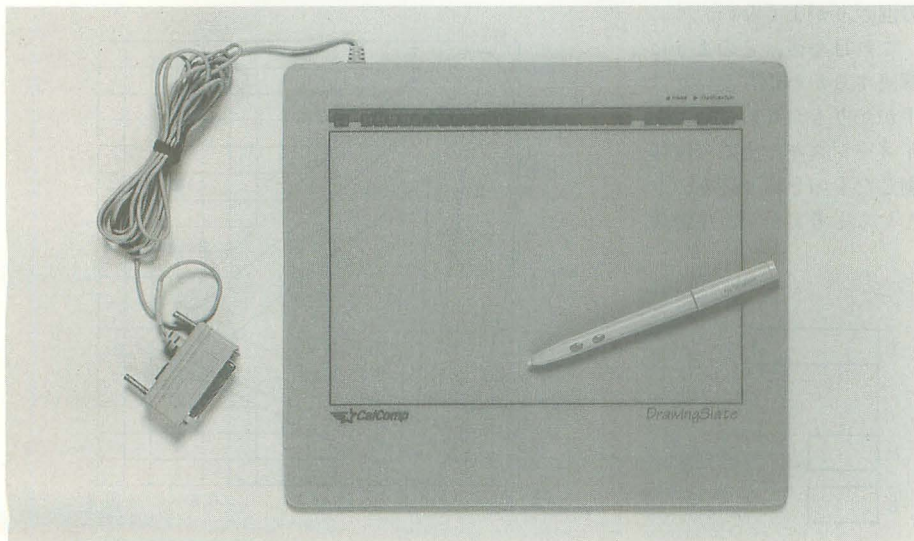
と入力するだけで常駐しますが，このときにタブレットを接続していないと弾かれてしまいますので，タブレットの準備をしてから実行してください。

また，/ms[n]というオプションをつけることで，マウスカーソルの移動速度を指定することができます。数値は1～3で指定し，数値が大きいほどマウスカーソルの移動速度が高速になります。このオプションはフリーウェアとして広く利用されている中村祐一氏のHIOCSのコンパチになっています。常駐を解除するには/rオプションをつけてください。

常駐すると，タブレットでマウスカーソルが制御できるようになります。マウスの右ボタンをクリックしてマウスカーソルを表示してから，そのカーソルがタブレットで操作できることを確認してください（マウスのON/OFFとソフトウェアキーボードの制御はタブレットではできません）。

マウスとの対応ですが，タブレットのボタン0（ペン先）はマウスの左ボタン，ボタン1は右ボタンに，ボタン2は右と左を同時にクリックすることに相当します。

また，このドライバは絶対モードにしか対応していませんが，メニューストリップをピックすることで，以下のような3種類の画面モードに対応させることができます。

15番	256×256
16番	512×512
17番	768×512

NSカルコンプ製Drawing Slate

起動直後は512×512モードになっています。また，メニューストリップの18番にはタブレットの初期化を割り当ててあります。動作に異常が生じた場合にピックしてください（メニューストリップを検知できないほど暴走した場合には無意味ですが）。

使用に際しての注意ですが，このドライバを常駐させたままタブレットを直接制御するアプリケーションや，RS-232Cを使用するものを起動しないでください。たとえマウスカーソルが表示されていなくとも，マウス割り込みが生きている限りタブレットが接続されているという前提でRS-232Cの入出力を行いますので，まず正常動作は望めません。

技術的なこと

X680x0のマウスの割り込みは，まずMFPのタイマC割り込みでマウス送信要求を出し，その後データを受信したSCCが再び割り込みをかけ，そこでボタンの情報や移動量などをIOCSのワークに格納するようになっています。ですから，ちゃんとしたマウス割り込み処理を記述するにはタイマCとSCCからの割り込みを根こそぎ書き直さなくてはならないのですが，今回はマウスの処理はそのままにタブレットの情報を重ねるという程度だったので，IOCSワーク$000938の指すマウス受信データ処理ルーチンMSCSRVECTと（これはSCC割り込みルーチンから呼ばれます），それに関連するいくつかのIOCSコールのみの変更にとどめました。

タブレットの送信要求およびデータ受信はMSCSRVECTから行い，マウスのデータとあわせてIOCSワークに格納しています。関連するIOCSワークの代表的なものとベクタを変更したIOCSコールを表1に示します。

ここで注意しなければならないことは，ROMのバージョンによってテキストVRAMへのマウスカーソルの表示方法が異なるということです。そこで初代～SUPERまでのROM　ver.1.0用とXVI以降のROM ver.1.1～ver.1.3用の関数を用意して，常駐時にROMのバージョンをチェックしてパッチを当てるようにしてあります。

タブレットの制御に関しては，付属のデモのソースをアセンブラで書き直した程度です。しかし，1回の割り込みでタブレットからRS-232Cで10バイトものデータを受信させると非常に重くなってしまうので，データがある分だけ受信し，続きはまた次回の割り込みで行うようにしてあります。

結果として，データがすべて揃うのに数回の割り込みを必要とすることになり，多少ぎこちなくなってしまいますが，やむをえないところでしょう。

また，ボタンに関してですが，16ボタンカーソルの場合は2つのボタンを同時に検知できないので，スタイラスペンと4ボタンカーソルでもボタン2でマウスの両ボタンクリックに対応させることにしました。ボタン0と1を同時に押した場合には，ボタン0が優先されます。画面モードはXSize, YSizeという変数で解像度を保持しています。

ただし，検知域をぴったり画面に対応させてしまうと，画面の端を指定しにくくなってしまうので，多少余裕をもたせるようにしてあります(図1)。また，検知域内で画面（クリッピング）外にカーソルがあるときには，はみ出した分をマウスの移動量としています。画面モードの変更はCalTab.sの1402行目からですので，余裕をもっと取りたいとか，違う画面モードに対応させたい場合などは各自で変更してみてください。

最後に

私はあまりアセンブラを使わないうえに，常駐モノを書いた（スケルトンは村田敏幸氏のですけど）のは初めてなので，それなりに動いてるだけでも結構驚異を感じてしまいます。たまにマウスカーソルが暴れたり，手抜きしてたりしますが，許してやってください。

このドライバの作成に際して，中村祐一氏からHIOCSのマウス割り込み部のソースの提供および指導をいただきました。また，タブレット制御部はDrawing Slateに付属のデモのソースを参考にさせていただきました。この場を借りてお礼を申し上げます。

表1　IOCSワークエリアの設定

IOCSワークエリア（詳しくは付録ディスク中のHIOCS.EQUを参照してください）

MSSKEYVECT	$000934.l	ソフトキーボードのマウス受信データ処理アドレス（今回は変更せず）
MSCSRVECT	$000938.l	マウスカーソルのマウス受信データ処理アドレス
MSLEFT	$000ac6.w	マウス左ボタンの状態（押されたら-1）
MSRIGHT	$000ac8.w	マウス右ボタンの状態（押されたら-1）
MSMOVEX	$000aca.w	マウスX方向移動量
MSMOVEY	$000acc.w	マウスY方向移動量
MSCSRX	$000ace.w	マウスカーソルX座標
MSCSRY	$000ad0.w	マウスカーソルY座標

ベクタを変更したIOCSコール

IOCS	$70	_MS_INIT
IOCS	$36	(_MS_VCS)

図1　検知域と画面の対応（512×512モードの場合）

BASICでマシン語を呼び出す

EXEC.FNC

Egawano Takashi 江川乃 誉司

高級言語すぎたためかX-BASICでは扱えなかったもの
それがマシン語レベルの操作関数です
EXEC.FNCはより自在な環境を作り出す最終兵器です

皆さん，お久しぶりです。前回は初めての原稿にもかかわらず，異様に態度のでかい文調でなんだか無性に恥ずかしくなってしまいました。というわけで，今回は頭ひとつ低い姿勢でいきたいと思います。

で，今回のお題はEXEC.FNCという代物です。FNCファイルということでX-BASICの外部関数だということはおわかりだと思いますが，これはX-BASICにマシン語環境を与えてくれるという関数群です。

ご存じのようにX-BASICにはCALLやPEEK，POKEといったようなハードに直接アプローチできる命令を持ちあわせていないため，BASIC上でマシン語プログラムを定義，実行することができません。とはいえ，マシン語をまったく利用できないわけではなく，FNCファイルとして登録することにより，外部関数として利用することができます。しかし，FNCファイルの作成は結構面倒な約束事や手続きなどが多く，ノーマニュアルで手軽に作成できるというわけではありません。また，汎用的な関数であればともかく，非常に用途の限られた非汎用的な関数をFNCとして登録するのはなんだか抵抗があるものです。

そこで，X-BASICでもX1や他機種のBASICのようにマシン語を扱えるような環境にしようではないか，ということで作成されたのがこのEXEC.FNCというわけです。

利用手順

利用手順ですが，まずはマシン語プログラムを作成しないことには始まりません。作成は普通にアセンブラを使用して行うのですが，X-BASIC上で動くプログラムということで，当然ある程度の規則が存在します。具体的な規則等は次の項で説明しますので，それに従ったプログラミングを行ってください。

作成したマシン語プログラムは，それをメモリに割り当てるわけですが，通常他機種の場合を見てみますと，必要量のメモリを確保し，そこへマシン語データをPOKE文で書き込んでいく，といった作業が必要となります。

それに対してEXEC.FNCの場合（というよりX-BASICの場合）は，単に1次元配列に定義するだけでそれら一連の作業をまかなえます。よって，ユーザーはメモリ確保といったことを考える必要がなく，当然POKE関数によるメモリ書き込みも必要ありません。配列定義の要領でずらずらとダンプリストを定義していくだけでOKです。とはいうものの，人間の手でそれを行うのはかなりしんどいことなので，ダンプリストをX-BASICの配列定義の形に変換してくれるツール「DLC.R」を用意しています。

このツールを使用すれば，ダンプリストをX-BASICに持ってくるのは非常に楽になるのですが，それでもアセンブルするたびに変換を必要としますので，まだまだ面倒な作業といえます。

そこで，もっと楽をする方法に，FREAD関数を使用する方法があります。つまり，実行ファイルを直接配列に読み出してしまうわけです。これなら，アセンブルしてもX-BASICのリストは書き換える必要がないため，効率よく開発できます。ですから，プログラムが開発段階のときはこの方法を使用し，前述のダンプリストを定義していく方法は，プログラムの完成後に1回だけ行えばよろしいでしょう。

こうして，配列に定義されたマシン語プログラムは，EXEC関数に放り込むことで実行できます。63ページにこのEXEC関数を含む各関数の使用法一覧を示しましたので，参照してください。

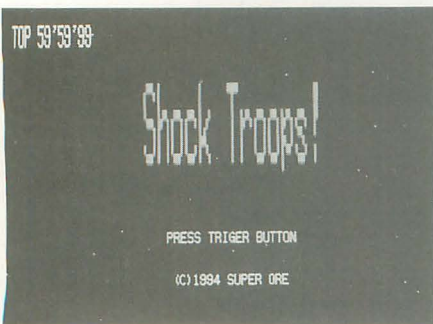
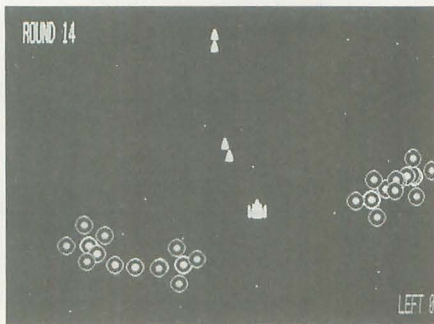
EXEC.FNC用プログラムの作成

EXEC.FNC用に作成されるマシン語プログラムは，単に高速なルーチンとして走らせるだけではなく，引数と戻り値を持つ関数として機能させることができます。そのため，X-BASICの外部関数用に作成されたものとほぼ同等のことが可能となっています。しかも，外部関数作成時の面倒な手続きなどは，ほとんどEXEC.FNCが受け持ってくれるため，ユーザーは純粋にプログラミングに専念できます。

さて，EXEC.FNC用のマシン語プログラムは，次の条件を満たしておく必要があります。

1）　実行ファイルが“*.R”形式のファイルであること。“*.X”形式のファイルでは動作しません。

2）　プログラム終了はRTS命令で行う。また，正常終了のときは必ずD0.Lに0をセットしてから終了する。

作成は以上の条件の下に行ってください。

サンプルゲームのタイトル画面

なかなか嫌らしい動きをする

次に，マシン語プログラム起動直後の，各レジスタの内容を示します。

A2 ＝パラメータ列のアドレス
A3 ＝システム変数のアドレス
A4 ＝プログラムの実行先頭アドレス
USP ＝親のスタックそのまま
SSP ＝OSのスタック

A2の指すアドレスには，EXEC関数から受け渡されるパラメータ列が，図1のような構造で格納されています。

63ページのEXEC関数の使用法のところを見てもらえばわかると思いますが，n1からn9までがユーザーに開放されたパラメータとなっています。これはどれもが省略可能で，各パラメータの有無に応じて，－2(A2).Wのビット状態が影響を受けます。パラメータ1から9までがそれぞれビット0から8に対応しており，パラメータが存在するところにビットが立つようになっています。

A3の指すシステム変数とは，EXEC.FNCで用意されたユーザー開放のメモリ領域のことで，1Kバイト確保されています。特に用途は決められてませんが，_SVP関数とPEEK,POKE関数を併用することにより，BASICからのアクセスが可能なうえ，複数のマシン語プログラムでも共通して利用できるので，それぞれの共用データを扱うのに適しているといえるでしょう。

図2にシステム変数の構造を示しましたので，参照してください。－2（A3）.Wに，システムコンディションというのが見られると思いますが，ここのビット0～14はユーザー開放で，ビット単位での使用を目的としています。

残りのビット15はBASICプログラムの起動状態を示しています。このビットが立っている場合，つまりシステムコンディションが負の場合はBASICプログラムが実行中（RUNされている）ということを表しています。

BASICの実行状況など一見なんの用途もないように思われますが，これはマシン語プログラム側での割り込み処理を考慮して設けたものです。

たとえば，走査線割り込みなどを使用中にもかかわらず，BASICプログラムを中断させたりした場合，当然異常の発生が予想されます。そのため，そのような場合には割り込みを禁止しなければなりません。そこで，このビットの恩恵が得られるわけです。つまり，割り込みルーチン側でこのビットを調べて，ビットが立っていなければ割り込みを禁止させることが可能となるわけです。

なお，システム変数およびシステムコンディションは，RUN命令実行時に初期化されます。

D0.L ＝エラーコード
D1.L ＝戻り値
A1 ＝エラーメッセージ格納アドレス

X-BASICへのデータの受け渡しは，プログラム終了時に以上のレジスタを介して行われます。

戻り値はint型固定となっており，受け渡しはD1.Lを使用します。

また，EXEC.FNC用のマシン語プログラムでは，X-BASICへのエラー出力も可能となっています。D0.Lの内容が0以外のときにエラーと判断され，このときD0.Lの内容が正の値であれば，A1で指すアドレスのエラーメッセージ（ヌル文字終了）を出力してくれます。D0.Lが負の値であれば，DOSコールのエラーコードに対応するエラーメッセージが出力されます。DOSコールのエラーコードについては，CコンパイラPRO-68Kのプログラマーズマニュアルをご覧ください。

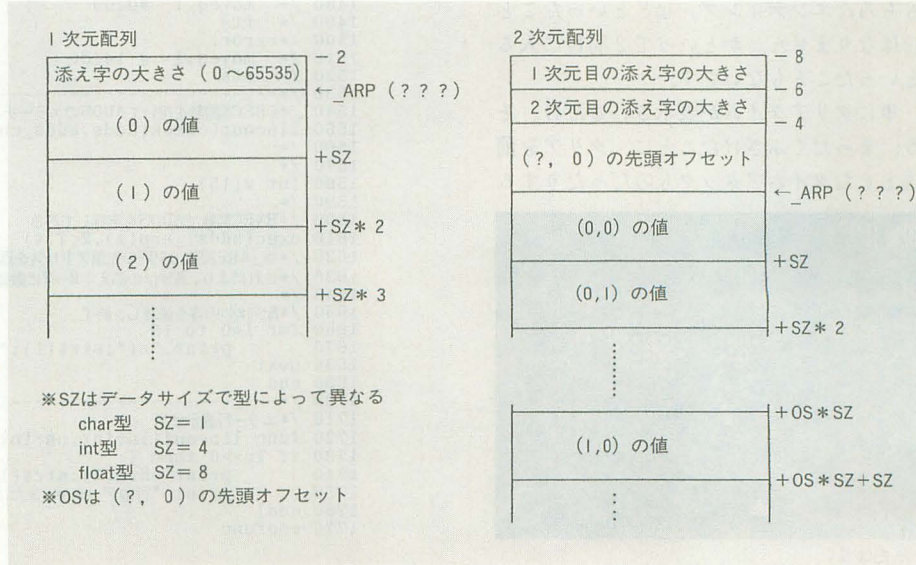
リスト1にEXEC.FNC用のマシン語プログラム作成例を示します。BASIC部も一緒に紹介していますので，各関数の使用例としても参考にしてください。

DLC.R について

前述のとおり，DLC.Rはダンプリストを X-BASICの配列定義の形に変換してくれるツールです。

次のように使用します。

DLC ［スイッチ］ ［ファイル名］

図3 配列の構造

変換後のファイルは行番号なしテキストファイルとなり，ファイル名は“＊.ary”のかたちとなります。

/Sスイッチを使用すれば，チェックサムデータを付加することができ，CHECK関数によるエラーチェックを行えるようになります。

SHOCK TROOPS! で遊ぼう

で今回，EXEC.FNCの紹介ついでに，サンプルとしてシューティングを作ってみました。タイトルの「SHOCK TROOPS！」ってのは突撃部隊の意を表しているんですけど，その名のとおり敵は弾を撃たずに，ひたすら突攻してくるという変なゲームに

図1 パラメータ列の構造

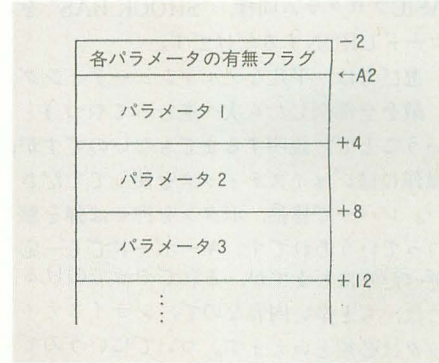
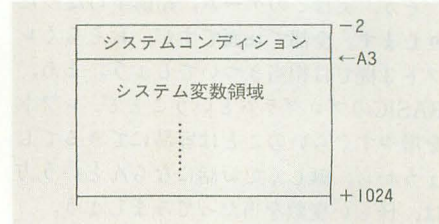
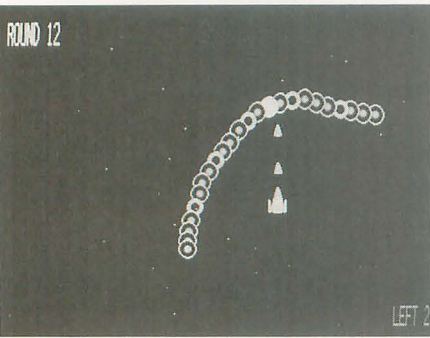



図2 システム変数の構造

頭を集中して狙え

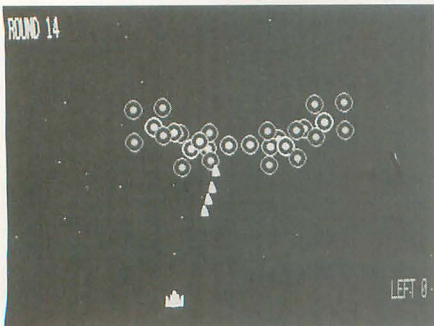
仕上がってます。

動作には，もちろんX-BASICにEXEC.FNCを組み込んでおく必要があります。また，10MHzユーザーの方は音楽ドライバ類のものははずしておいたほうがよいです。処理落ちがなくなります。あとは普通のB-ASICプログラム同様，“SHOCK.BAS”をロードしRUNするだけです。

遊び方は，平凡なノルマシューティング（敵を全部倒したら次へ進むってやつ）ということで，説明するまでもないのですが，操作にはジョイスティックを使ってください。レバーで移動，ボタンを押せば弾を撃つっていうあれです。キーボードでも一応遊べたりはしますが，それでやってのけるには，ちと辛い内容なので，ジョイスティックは必須といえます。ついでにいうのであれば，連射機能つきのやつ。

そう，実はこのゲーム，結構辛口だったりします。全部で18面ですが，おそらくレフト3機では相当きついでしょう。まあ，BASICのプログラムということで，レフトを増やすぐらいのことは容易にできるでしょうから，難しくてお話にならんという方は，怪しい変数を当たってみましょう。

で，この超難度の中を無事切り抜けると，もちろんエンディング。などといったことにはなりません。かといって2周目に入るといったこともないです。

単にクリアタイムが表示されるだけ。そう，まったくふざけたことに，クリアを前提としたタイムアタックものだったりするのです。

この超難度だというのに，本当にふざけていますが，すべて時間がなかったことが原因です。時間があれば難易度調整をもっと行ったんですけど，最後のほうはそれよりも面数を増やすことに気をとられて，そっちはまったく手つかずでした。

しかも，ここにきてバグが見つかりました。大バグというわけではないのですが，たまに10面，14面で全部破壊したにもかかわらず，次に進めなくなることがあります。正確には敵が画面外に出たまま戻ってこなくなり，倒しようがなくなります。

最初は，まあ滅多に起こらないからいいかぁ，なんて思っていたのですが，この超難度とともに気になってしょうがないので，パッチを作ることにしました。リスト2がそれです。ディスクつきの号だというのに，余分な手間をかけさせて本当に申しわけないのですが，付録ディスクの“SHOCK.

まだ先は長い

リスト1　EXEC.FNC用 マシン語プログラム例

```
1000 /*EXEC.FNC用 マシン語プログラム例 ADDS
1010 /*
1020 /*exec(adds,_na,an1,an2,n)
1030 /*パラメータ
1040 /*     _na  = 1次元配列のアドレス
1050 /*     an1  = 添え字
1060 /*     an2  = 添え字
1070 /*       n  = 加算数値
1080 /*
1090 /*機能
1100 /*     _naで指定した1次元配列の添え字an1からan2までに
1110 /*     数値nを加算し代入する
1120 /*
1130 /*ADDSのダンプリストを配列に定義
1140 int adds(14)={
1150 &HC6A000F ,&HFFFE6630,&H2C5A7E00,&H3E2EFFFE,&H2C1A2A1A,
1160 &H2812BC87,&H621EBA87,&H621A3E05,&H9E466404,&H44473C05,
1170 &HDC86DC86,&HDDC6D99E,&H51CFFFFC,&H70004E75,&H70F24E75}
1180 /*
1190 /*ADDSのエラーチェック用データ
1200 int adds_chk(0)={
1210 &H4BE406D }
1220 /*-----------------------------------------------------------
1230 /*ADDSのアセンブリソース
1240 /*  cmpi.w   #%1111,-2(a2)  パラメータ使用フラグ
1250 /*  bne      error          パラメータが不適合
1260 /*  move.l   (a2)+,a6       パラメータ1  配列ポインタ
1270 /*  moveq.l  #0,d7
1280 /*  move.w   -2(a6),d7      配列の最大添字
1290 /*  move.l   (a2)+,d6       パラメータ2  添字1
1300 /*  move.l   (a2)+,d5       パラメータ3  添字2
1310 /*  move.l   (a2),d4        パラメータ4  加算数
1320 /*  cmp.l    d7,d6
1330 /*  bhi      error          配列の最大添字を越えている
1340 /*  cmp.l    d7,d5
1350 /*  bhi      error          配列の最大添字を越えている
1360 /*  move.w   d5,d7
1370 /*  sub.w    d6,d7          ループ回数
1380 /*  bcc      offsetadd
1390 /*  neg.w    d7
1400 /*  move.w   d5,d6
1410 /*offsetadd:
1420 /*  add.l    d6,d6
1430 /*  add.l    d6,d6
1440 /*  adda.l   d6,a6          作業開始アドレス
1450 /*loop:
1460 /*  add.l    d4,(a6)+
1470 /*  dbra     d7,loop
1480 /*  moveq.l  #0,d0          エラーなし
1490 /*  rts
1500 /*error:
1510 /*  moveq.l  #-14,d0        エラー終了
1520 /*  rts
1530 /*-----------------------------------------------------------
1540 /*CHECK関数を用いてADDSのエラーチェックを行う
1550 lineout(check(adds,adds_chk),1150)
1560 /*                                  ↑ダンプリストの先頭行
1570 /*
1580 int z(15)
1590 /*
1600 /*EXEC関数でADDSを実行してみる
1610 exec(adds,_arp(z),2,7,4)
1620 /*※_ARP関数は配列の先頭アドレスを返す関数
1630 /*これにより、配列zの添え字2～7に数値4が加算される
1640 /*
1650 /*配列zの内容を確認し、終了
1660 for i=0 to 15
1670         print "z(";str$(i);")";chr$(9)"=";z(i)
1680 next
1690 end
1700 /*-----------------------------------------------------------
1710 /*エラー行表示サブ
1720 func lineout(ln;int,os;int)
1730 if ln>=0 then {
1740         print chr$(7);str$(ln*10+os);
1750         print "行のデータか、または対になるチェックデータに誤りがあります"
1760 end}
1770 endfunc
```

BAS”に，このパッチを当ててください。

このパッチを追加すると，自機が画面端に行ってもスクロールストップしなくなり，もっと先に行けるようになります。つまり，画面外から帰ってこないバカ野郎も破壊することができるわけです。

また，14面と最終面の難易度が下がります。それ以外の面は変わりませんが，それで結構なんとかなるものです（12面は少々くせものですが)。私はシューティングのプロというわけではありませんが，これでノーミスクリアも可能となります。自称シューティングプロの皆さんは諦めずにノーミスクリアを目指しましょう。

さて，タイムアタックというものはクリアが比較的容易になってからが始まりというものです。このパッチによって，なんとかそのお膳立てができました。ということで，クリアできるようになったらタイムアタックに勤しんでください。

まず初めの相場としては，5分台後半といったところでしょうか。さらにやり込めば，4分台も出せるはずです。しかし，本当の勝負は……。

最後に注意点を思い出したので，つけ加えておきます。このゲームではPOKE関数を使用することにより，384×256の画面を実現しています。そのためプレイ後，ほかのアプリケーションに移ると画面が乱れることがあります。これを防ぐためには，ほかのアプリケーションに移る前に，BASIC上で［CTRL］+［D］を行ってください。

最後に

今回サンプルにシューティングを作ったわけですが，考えてみればマシン語（といってもBASICで動くのだが）でゲームを作るのはこれが初めてなんです。別に言い訳してるわけじゃないけど，かなり手抜きの多い変なゲームになってしまいました。でも，敵のアルゴリズムは結構気にいってたりします。特に形状記憶蛇なんかいいと思うのですが，いかがなもんでしょうか。よかったら感想を聞かせてください。

ところで，前回紹介したICEですが，あのあとなんの進展もないまま現在に至っております。ですから，いまごろはきっとゴミと化しているんじゃないでしょうか。

そこで，次はICEをどうにかしようと考えています。ライブラリや周辺でのサポート体制を整えて，さらに圧縮比をPICの半分ぐらいにまで高めて，圧縮率も速度もPIC2を上回るものにしたいところです。すでに頭の中では，それなりのアルゴリズムができ上がっているのですが，いざ実行してみるとダメっていう可能性もあるから，あまり期待しないで待っててください。

それでは次はいつになるかわからないけど，そんときまでのさらばだ！

リスト2 SHOCK TROOPS! デバッグ&難易度変更パッチ

```
3115 patchsub()
4980                      /*----- デバグ&難易変更パッチ
4990 func patchsub()
5000 int patch(33)={
5010 &H2C5A2E3C,&H4E714E71,&H2D470114,&H2D470126,&H2D470148,
5020 &H2D47015C,&H2C52532E,&HC817038 ,&H2200E549,&H2A4E284E,
5030 &H7E033C3C,&H1E0552D ,&H292B592D,&H2A0B542D,&H2B35552D,
5040 &H2C3B592D,&H2D1B542D,&H2E45DAC0,&H394637F2,&H397C03C0,
5050 &H37F4DC7C,&H3CD8C1  ,&H51CFFFD4,&H1D7C0028,&H3D261D7C,
5060 &H283D2A  ,&H7E071D7C,&HA3B53   ,&H522E3B65,&H522E3B69,
5070 &H66E0168 ,&H3B72DCC0,&H51CFFFE8,&H70004E75}
5080 int patch_chk(1)={
5090 &H44340BA ,&H5DFCCA4B}
5100 lineout(check(patch,patch_chk),5010)
5110 exec(patch,_arp(argset),_arp(enemy))
5120 endfunc
```

DLC.R使用の実際

実際にDLC.Xを使ってみましょう。

付録ディスクについているE.Xは「Bドライブをイジェクトする」というまったく低機能なコマンドですが，これを＊.Rファイルにコンバートして X-BASICから呼び出してみました。

CV.Xでコンバートしたものを，

```
A>DLC -S E.R
```

で出力すると，

```
int e(5)= {
&HFF0D3F3C,&H302FF0F ,&H548F3F3C,&H102FF0F ,&H548FFF00,
&H4E714E71}
int e_chk(0)= {
&H484F400 }
```

のようなものが得られます。

このプログラムはDOS_EXITで終了していますので，ちょっと書き換えましょう。DOS_EXITは“FF00”というコードです。これをRTSを表す“4E75”に書き換えます。

このまま，

```
exec(e)
```

を行うと外部関数エラーが出てしまいます。ディスクイジェクトを行うDRVCTRLは返り値を持たないからです。幸い，後ろについている2つの“4E71”はNOPですのでここも書き換えて最終的に，

```
10 int e(5)= {
20 &HFF0D3F3C,&H302FF0F ,&H548F3F3C,&H102FF0F ,&H548F7000,
30 &H4E754E71}
40 int e_chk(0)= {
50 &H484F400 }
60 exec(e)
```

のようになります。

EXEC.FNC 各関数の使用法一覧

_ARP

書式 _arp (na)

機能 naで指定された配列の先頭アドレスを返します。naは配列であれば型，次元数は問いません。配列の構造については図3を参照してください。

_SVP

書式 _svp ([n])

機能 nで指定された値をオフセットとしてシステム変数のアドレスを返します。nは0～1023までの値を取ることができます。nを省略した場合はシステムコンディションのアドレスを返します。

システム変数については本文および，図2を参照してください。

PEEK (PEEKB,PEEKW,PEEKL)

書式 peek (_adr)

機能 _adrで指定されたアドレスの内容をワード単位で返します。

なお，PEEKBはバイト単位，PEEKLはロングワード単位となります。

POKE (POKEB,POKEW,POKEL)

書式 poke (_adr,n1 [, n2,n3 …… , n9])

機能 _adrで指定されたアドレスに，続くパラメータ列の内容をワード単位で書き込みます。パラメータは一度に9個まで書き込むことができます。

なお，POKEBはバイト単位，POKELはロングワード単位となります。

MOVE

書式 move (_adr1, _adr2,lg)

機能 _adr1で指定されたアドレスからlgバイト分の内容を_adr2に転送します。

EXEC

書式 exec (na [, n1] [, n2] [, n3] …… [, n9])

機能 naで指定された配列の内容を，マシン語プログラムとして実行します。naは1次元配列で型は問いません。

n0～n8はint型のパラメータでユーザーが自由に利用できます。これらのパラメータの受け渡しについては，本文および図1を参照してください。

CHECK

書式 check (na,cd)

機能 naで指定された配列の内容をcdのチェックサムデータによってチェックします。

エラーを発見した場合は，naの定義開始行から何行目にエラーがあったのかを返します。エラーがなかった場合は－1を返します。

cdのチェックサムデータはDLC.Rによって作成されます。DLC.Rについては本文を参照してください。

高度なスプライト操作を手軽に

XSPRITE.FNC

Ito Masahiko 伊藤 雅彦

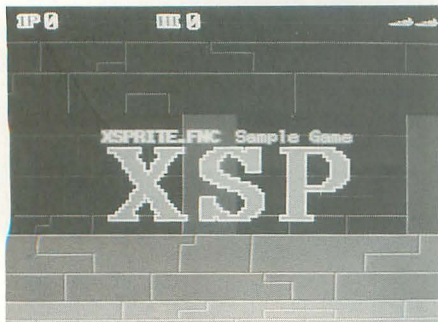
X68000のもつスプライト表示能力をBASICレベルで引き出します
XSPRITEはお決まりの処理を高速にサポートします
インタプリタ上で有効なほか，コンパイルにも対応しています

XSPRITE.FNCのご紹介

X68000でゲームを作ろうというとき，ありがたいのがスプライト機能です。たくさんのキャラクターをすいすい動かすのもスプライトを使えば簡単です。

んが，X-BASICを使うと，たくさんのキャラクターをすいすい，というわけにはいかなくなってしまいます。実行スピードが遅いので，スプライトの力を持ってしてもドット単位で複数のキャラ移動をすると超スロー映像になってしまいます。しかたなく8ドットぐらいの単位で移動させてみても，動きがガタガタして，これでもスプライトなのかと涙を落とすことになります。

X-BASICでも，スプライトをもうちょっとスプライトらしく扱えないものでしょうか。そりゃ，XBAStoCを使えばどうにでもなりますけど，インタプリタ上でちゃんと動いてくれたほうがデバッグしやすくて，ずっと気軽にプログラムが作れるでしょう。

ということで，X-BASICでスプライトが飛び交うゲームが作れるようにと作ってみた外部関数がXSPRITE.FNCです。この関数を使ってできることというのは，スプライトのスライド移動です。ある座標に表示されているスプライトに対して，移動先の座標と移動スピードを与えると，スプライトが移動先までずりずりとスライドしていく，というものです。もちろん，スライドの処理は割り込みでやって，スライドさせている最中でもプログラムの実行は止まらないようにしています。

サンプルゲームだ

また，そのほかにゲーム作りに役立ちそうな機能も盛り込んでみました。やっていることは簡単なんですが，これでもX-BASICの可能性はかなり広がると思います。それに，私のプログラミング力からいって，これぐらいのものを作るのがちょうどよかったんです(68000でプログラムを組むのは初めてなもので)。

組み込み方法

XSPRITE.FNCの関数を使うためには，エディタを使ってBASIC.CNFに，

FUNC=XSPRITE

という1行を加えて，BASIC.CNFと同じディレクトリにXSPRITE.FNCをコピーします。これでX-BASICを起動すれば，今回の関数が組み込まれます。

関数の概要

XSPRITE.FNCには24個の関数が詰め込んであります。その詳しい説明は来月にさせていただくとして，ここでは概要だけ紹介しておきます。

●スプライトのスライド

sp_slidep (s, [x], [y], v)
sp_slidev (s, vx, vy, t)

スプライトのスライドを実行するのが，この関数です。2つあるのは，パラメータの指定方法を2種類用意したからです。sp slidep ()では移動先の座標と移動スピードを指定し，sp_slidev () では移動ベクトルと移動時間を指定します。

●スプライトの移動

s_ploc (s, [x], [y])

スプライトを指定座標に移動します。これは既存のsp_move () やsp_set () でもできることですが，XSPRITE.FNCの機能を使っている場合には必ずこのsp_loc()を使って移動するようにしてください。

●スプライトの連結

sp_hang (s1, s2, x, y)
sp_leave (s)

スプライトの大きさは16×16ドット。これではちょっと小さいということも，ままあるでしょう。そこで，2つ以上のスプライトを連結して，ひとつのスプライトのようにスライド・移動をさせることができるようになっています。

●接触判定

sp_hgadd (g, s)
sp_hgrmv (g, s)
sp_hitrng (s, x1, y1, x2, y2)
sp_hiton (s, g)
sp_hitoff (s)
sp_hit (s)

ゲームを作るなら絶対欲しい，スプライトの接触判定です。接触判定関係は使い方がちょっと複雑になってしまいました。

たとえば，スプライト1とスプライト2〜4との接触判定をするときには，まず，

sp_hgadd (0,2)
sp_hgadd (0,3)
sp_hgadd (0,4)

と，接触判定グループ0にスプライト2〜4を登録しておいてから，

sp_hiton (1,0)

で接触判定を始めます。その後，sp_hit(0)の戻り値をチェックすることで接触状況を知ることができます。

注意してほしいのは，sp_hit ()の戻り値はリアルタイムの接触判定値ではないということです。戻り値の意味は，「いま接触しているか」ではなくて，「接触判定を始めるか，または前回の接触状況をチェックしてから(つまり，sp_hiton () かsp_hit () を実行してから)，現在までに接触したことが

あるか」なのです。どうしてこうなっているかというと，スライド中のスプライトが一瞬だけほかのスプライトに接触した場合，リアルタイムに値をBASICに返すと，接触したという情報を取り逃がすことがあるからです。

それから，各スプライトの接触判定の範囲は，sp_hitrng () で任意の矩形領域に変えることができます。

●ジョイスティック操作

sp_stkon (j , s, vx , vy)

sp_stkoff (j)

たいていのゲームでは，マイキャラはジョイスティックを上に倒せば上に動き，右に倒せば右に動くようになっています。そこで，スプライトをジョイスティック操作に応じてスライドするようにしてしまう関数も用意しました。

●バックグラウンドのスライド

bg_slide (b , vx , vy , t)

BGもスライドさせることができます。BGの場合はスクロールといったほうがいいんでしょうか。スムーズな背景スクロールができます。

●マップスクロール

bg_map (b , x , y , ca)

bg_mpscrl (b, [x], [y])

BGのスライドに関しては，通常モードとマップモードがあって，マップモードではBGの領域を仮想的に64×64より大きく(または小さく)することができます。情けないことに完全に実BGと同じように扱うことはできないんですが(BGの書き換えができないんです)，スクロールゲームでは大きい背景を用意することができるので便利でしょう。

●その他の関数

XSPRITE.FNCの関数を使うときには，まず初期化の関数sp_xinit ()を実行してください。既存のスプライト初期化関数sp init () は，sp_xinit () の前に実行しないといけないので，結局のところ，スプライトの初期化の処理は，

sp_init () :sp_xinit ()

となります。

また，割り込み処理を開始するにはsp slon(t)を実行します。tは割り込み周期で，普通は0を指定して垂直帰線期間に入るたびに割り込みがかかるようにします。スライドや接触判定の処理は割り込みでやっていますから，sp_slon () を実行しないとなんにも起こりません。また，プログラム終了時にはsp_sloff ()で割り込みを止めるようにしてください。

その他の関数については来月に説明しますが，関数リファレンスのほうも参照してみてください。

サンプルゲーム

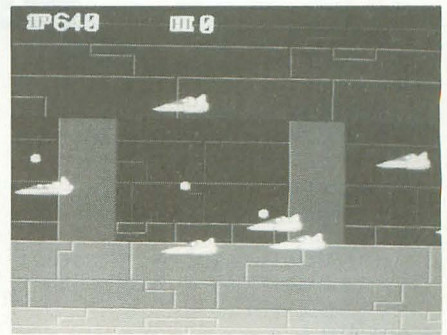
XSPRITE.FNCがどういうものか，目で見てわかってもらおうと，サンプルを作りました。一応，横スクロールシューティングの形になっています。操作はジョイスティックの上下左右で自機の移動，トリガーAで弾の発射，タイトル画面でトリガーBを押すとゲーム終了となっています。ちなみに，「敵要塞に侵入していた偵察機が任務終了直後に敵に発見され，敵機の追撃を振り切って要塞脱出を図る」という，どうでもいいストーリー設定があります。

では，XSPRITE.FNCを組み込んだX-BASICを起動して，xsp.basを実行してみてください。

ゲーム自体は敵機をえんえんと撃ち墜とすだけで，いつまでたってもなんの新展開もない退屈なものです。んが，弾の連射や墜落機から出る煙などでスプライトをふんだんに使いながら，このキャラクターのスムーズな動き，そして背景の2重スクロールなど，BASICとは思えない画面に「おおっ」と思う……のは030ユーザーだけかも。10MHz機では弾の連射が利かない，撃墜時の煙もポコッポコッと出るだけという状態になってしまいます。しかし，これは10MHz機ではXSPRITE.FNCでたくさんのスプライトを処理する能力がないということではありません。BASICのメインループが遅くて，弾や煙を「出す」処理が遅くなってしまうためで，弾や煙を「動かす」処理をする余力はまだあります。ま，なんにしてもこのサンプルが10MHz機では無理があるのは確かなので，10MHz機のほうはプログラム中の変数clockを10に変更して実行してください。

さて，このプログラムのメインループでやっている処理は，

1）自弾の発射
2）敵機の移動と敵弾の発射
3）煙の発生
4）自機と敵機・敵弾の接触チェック
5）自弾と敵機の接触チェック

といったところです。面白いのは，自機の移動処理が全然ないということです。スプライトをジョイスティック操作によってスライドさせる関数sp_stkon () をメインループ前に置いておけば，自機の移動は勝手にやってくれるわけです。また，ほかのキャラの移動も「あっちへスライドしろ」と指令しているだけです。あとは接触チェックやスライドが終わったかどうかのチェックなんかをやっています。

後ろへ向けて弾を撃つ

XSPRITE.FNC関数を使ったプログラムは，だいたいこのサンプルのようにスプライトの状態を監視しながらスライドの指令を出していく，というパターンになると思います。

Cへのコンバート

XSPRITE.FNCの関数を使ったBASICプログラムは，XBAStoCを使ってCにコンバートすることができます。この関数はもともとインタプリタでもスプライトをばんばん使えるようにと作ったものですので，コンパイルするのは本来の目的からははずれているんですが，実際にはそれでもコンパイルしたいときというのはあるでしょう。

BCコンバータ関連のファイルはBtoC kitというディレクトリに入っています。XSPRITE.DEFはBCディレクトリに，XSPRITE.HはINCLUDEディレクトリにコピーして，*.oファイルはBASLIB.Lに登録してください。

また，サンプルゲームxsp.basをコンパイルしたxsp.xも収録しておきましたので，試しに実行してみてください。コンパイルすると，10MHz機でも連射ばしばし，煙もくもくになります。

＊　＊　＊

来月にはもっと詳しく関数の使い方を説明しますが，自分でいろいろ試してみるだけでも使い方はわかると思います。ちょっといじってみてください。

サンプルではごくありきたりのシューティングゲームなんか作ってしまいましたが，本当はアイデア一発のピコピコゲームっぽいものがいろいろ作れるんじゃないかと期待をかけています。スプライトがスムーズに動くX68000らしいピコピコゲーム，あなたも作ってみませんか。

XSPRITE.FNC関数リファレンス

sp_xinit ()
戻り値:なし
XSPRITE.FNC関数の初期化を行う。スプライトの初期化の際には，sp_init ()実行後に当関数を実行する。

sp_slon (t)
t : [char,0～15] 割り込み周期
戻り値:なし
垂直同期割り込みを開始する。引数の値により，垂直同期周期(t+1)回に1度の周期で割り込み処理を行う。

sp_sloff ()
戻り値:なし
垂直同期割り込みを停止する。

sp_slidep (s, [x], [y], v)
s : [char,0～127] スプライトのプレーン番号
x : [int,0～1023] スライド先のX座標(省略時は現在のX座標)
y : [int,0～1023] スライド先のY座標(省略時は現在のY座標)
v : [int,1～2047] 移動スピード
戻り値:なし
スプライトsを座標(x,y)にスライドさせる。vはX方向・Y方向のうち，移動幅の大きいほうの移動スピード。

sp_slidev (s,vx,vy,t)
s : [char,0～127] スプライトのプレーン番号
vx: [int,-2047～2047] X方向の移動ベクトル(右方向が+)
vy: [int,-2047～2047] Y方向の移動ベクトル(下方向が+)
t : [int,1～32767] スライド時間
戻り値:なし
スプライトsを指定したベクトルでスライドさせる。引数tにより，(t×[割り込み周期])の間だけスライドする。たとえばsp_slon (0)が実行された状態でtに120を指定すると，120×(1/60)=2秒間スライドする。

sp_loc (s, [x], [y])
s : [char,0～127] スプライトのプレーン番号
x : [int,0～1023] X座標(省略時は現在のX座標)
y : [int,0～1023] Y座標(省略時は現在のY座標)
戻り値:なし
スプライトsを座標(x,y)に表示する。

sp_hang (s1,s2,x,y)
s1:[char,0～127] スプライトのプレーン番号
s2:[char,0～127] スプライトのプレーン番号
x :[int,-128～127] 連結位置X
y :[int,-128～127] 連結位置Y
戻り値:なし
スプライトs1にスプライトs2を連結する。連結する位置はスプライトs1の位置を(0,0)としてドット単位で指定する。3個以上のスプライトをいもづる式に連結することもできる。

sp_leave (s)
s :[char,0～127] スプライトのプレーン番号
戻り値:なし
プレーン番号sのスプライトに連結したスプライトを切り離す。

sp_hgadd (g,s)
g:[char,0～15] 接触判定グループ番号
s:[char,0～127] スプライトのプレーン番号
戻り値:なし
スプライトsを接触判定グループgのメンバーに加える。接触判定グループとは，sp_hiton()関数でスプライトの接触判定を行う際に，どのスプライトとの接触を判定するかを指定するためのものである。

sp_hgrmv (g,s)
g:[char,0～15] 接触判定グループ番号
s:[char,0～127] スプライトのプレーン番号
戻り値:なし
スプライトsを接触判定グループgのメンバーから削除する。

sp_hitrng (s,x1,y1,x2,y2)
s:[char,0～127] スプライトのプレーン番号
x1:[char,0～255] 接触判定範囲左上のX座標
y1:[char,0～255] 接触判定範囲左上のY座標
x2:[char,0～255] 接触判定範囲右下のX座標
y2:[char,0～255] 接触判定範囲右下のY座標
戻り値:なし
スプライトsの接触判定範囲を設定する。

sp_hiton (s,g)
s:[char,0～127] スプライトのプレーン番号
g:[char,0～15] 接触判定グループ番号
戻り値:なし
スプライトsの接触判定を開始する。判定を行うのは接触判定グループgのメンバーになっているスプライトとスプライトsとの間の接触だけである。
接触判定は割り込み処理中に行われ，接触があった場合にはsp_hit (s)の戻り値を接触したスプライトのプレーン番号にして接触判定を停止する。接触がなかった場合には次回の割り込み処理でも判定を続ける。

sp_hitoff (s)
s:[char,0～127] スプライトのプレーン番号
戻り値:なし
スプライトsの接触判定を停止する。

sp_hit (s)
s:[char,0～127] スプライトのプレーン番号
戻り値:[int,0～127 or 255]
接触したスプライトのプレーン番号(255=接触なし)
スプライトsの接触判定状況を返す。当関数は一度実行すると次回からの戻り値が255にリセットされる。

sp_stkon (j,s,vx,vy)
j :[char,1～2] ジョイスティック番号
s :[char,0～127] スプライトのプレーン番号
vx :[char,0～255] X方向の移動ドット数
vy :[char,0～255] Y方向の移動ドット数
戻り値:なし
スプライトsをジョイスティックjの操作に応じてスライドするようにする。
移動可能範囲はグラフィック画面で設定する。512×512ドットモード画面(16,256,65536色のどのモードでもよい)のスクリーン0に，移動可能にしたい座標にだけ奇数のパレットコードを書き込む。

sp_stkoff (j)
j:[char,1～2] ジョイスティック番号
戻り値:なし
sp_stkon ()関数で設定したジョイスティック操作によるスライドを中止する。

sp_xstat (s)
s:[char,0～127] スプライトのプレーン番号
戻り値:[int,0～7] スライド・連結状況
スプライトsのスライド・連結状況を返す。
bit 0: 1=スライド中
　　0=スライド中ではない
bit 1: 1=他のスプライトに連結している
　　0=他のスプライトに連結していない
bit 2: 1=ジョイスティック操作中
　　0=ジョイスティック操作中ではない

sp_intcnt ()
戻り値:[int,0～4294967296] 割り込み回数
sp_xinit ()を実行してから現在までの垂直同期割り込みの割り込み回数を返す。

bg_slide (b,vx,vy,t)
b:[char,0～1] バックグラウンド番号
vx:[int,-2047～2047] X方向の移動ベクトル(左方向が+)
vy:[int,-2047～2047] Y方向の移動ベクトル(上方向が+)
t:[int,0～32767] スライド時間
戻り値:なし
バックグラウンドbを指定したベクトルでスライドさせる。引数tにより，(t×[割り込み周期])の間だけスライドする。tに0を指定した場合は時間制限なくいつまでもスライドする。また，スライドを停止する場合はvxとvyの両方に0を指定する。

bg_map (b,x,y,ca)
b:[char,0～1] バックグラウンド番号
x:[int,33～32767 or 0] 仮想画面サイズX
y:[int,33～32767 or 0] 仮想画面サイズY
ca:[char型2次元配列] マップデータ
戻り値:なし
バックグラウンドbをマップモードにする。マップモードでは任意の大きさのバックグラウンド面を仮想し，仮想画面の表示，スライドを行うことができる。
仮想画面の内容(マップデータ)はchar型2次元配列caで指定する。caの0,1番目の要素に仮想座標(0,0)のパターンコードを代入し，以下(1,0)，(2,0)，…(x-1,0)，(0,1)，(1,1)，…(x-1,y-1)のパターンコードを代入しておけばよい(したがってcaの要素数はx×y×2)。通常は，
dim　mapdata (x×y-1,1)
と配列宣言し，mapdata (?, 0)に反転指定とパレットブロック，mapdata (?, 1)にパターンコードを代入するとよい。
また，x,yともに0を指定して当関数を実行すると，マップモードが解除される。

bg_mpscrl (b, [x], [y])
b:[char,0～1] バックグラウンド番号
x:[int,0～] X座標(省略時は現在のX座標)
y:[int,0～] Y座標(省略時は現在のY座標)
戻り値:なし
マップモード時，仮想画面の座標で表示位置の設定を行う。

bg_xs_tat (b,md)
b :[char,0～1] バックグラウンド番号
md:[char,0～2] モード
戻り値:[int,0～1] 各種スライド状況
バックグラウンドbのスライド状況を返す。
md=0のとき
　bit0: 1=スライド中
　　0=スライド中ではない
　bit1: 1=マップモード
　　0=通常モード
md=1のとき
マップモード時，画面の左上隅のチップの仮想X座標
md=2のとき
マップモード時，画面の左上隅のチップの仮想Y座標

xstrig (j,tr)
j:[char,1～2] ジョイスティック番号
tr:[char,0～1] トリガー指定(0=トリガーA,1=トリガーB)
戻り値:[int,0～4294967295] トリガー連打回数
前回この関数が実行されて以降，ジョイスティックのトリガーが押された回数を返す。

scrpri (sp,tp,gp)
sp:[char,0～2] スプライトの表示優先順位
tp:[char,0～2] テキストの表示優先順位
gp:[char,0～2] グラフィックの表示優先順位
戻り値:なし
スプライト・テキスト・グラフィック面の表示優先順位を設定する。引数には優先順位が高いほうから0,1,2を指定する。

それは耳から脳髄が溶け出すほどに……

PUSH BON! オリジナルステージ大集合

Takahashi Tetsushi 高橋 哲史

合計210ステージ集まった「PUSH BON!」オリジナルステージ
個性豊かなステージたちに高橋氏もずいぶん苦しめられたようです
皆さんも付録ディスクから解凍して，秋の夜長を「PUSH BON!」三昧してください

こいのぼりPRO-68Kで5年ぶりにリニューアルされて登場したパズルゲーム「PUSH BON!」。読者の皆様方にも好評のようで，微力ながらBGM書きで開発に協力させていただいた私も非常に嬉しいかぎりであります(BGMのほうは気に入っていただけたでしょうか?)。

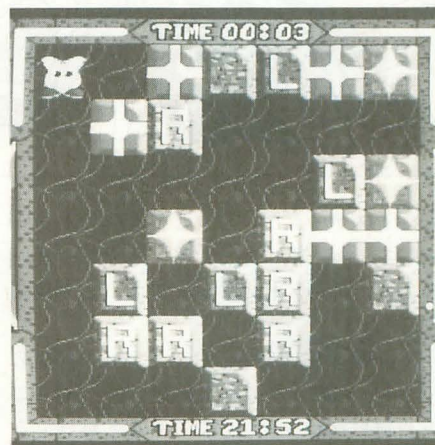
さて今回は，デフォルトの50ステージだけでは飽き足らず，自ら難面奇面を作成し投稿してくださった皆さんのオリジナルステージに私こと高橋哲史がチャレンジさせていただこう，とこういった趣向なわけでございます。「目指せ全ステージクリア！」を合言葉にやる気満々でアタックを開始したわけですが，さてその結果は？

疾風怒濤の100ステージ攻勢！

まずは読者投稿第1弾の周東さんの作品からいきましょう。なんと周東さんは9月号に掲載された50ステージに加えて，新たに50ステージ，合計100ステージ(！)ものオリジナルステージを作成してくださいました。しかも1ステージ1ステージに名前までついています(これがなかなか凝ってるんだ)。

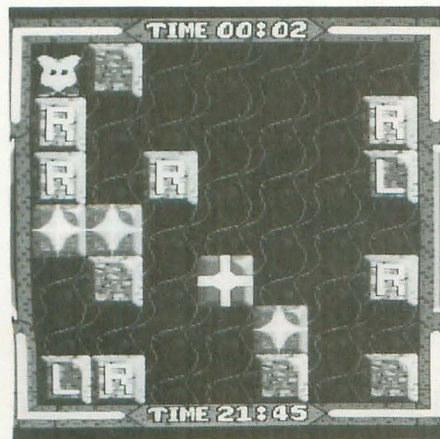
それでは9月号に掲載された前半50ステージ(周東さん曰く「〝静〟の50ステージ」だそうです)にチャレンジしてみます。結果から先に書いてしまうと，私に解けたのはたったの35ステージ。全体の7割だけでした。何度もトライしてみたのですが，どーしてもこれ以上解けません(泣)。

のっけから全ステージ制覇の夢がもろくも崩れさってしまいましたが，提示された最小ステップを下回るステップでクリアしたステージが4ステージありましたのでそれで勘弁していただきましょう。ちなみにそのステージとステップ数は，18ステージ11ステップ，35ステージ15ステップ，37ステージ5ステップ，41ステージ9ステップです。皆さんもどんどん最小ステップ破りをして作者の周東さんをいじめてあげましょう。

最初にも少し書きましたが，周東さんの各ステージごとのネーミングはなかなか光るものがあります。YMOやリトルスターなど見たまんまのステージ名もあるのですが，解き方を暗示したマイマイ，エレベーター，あみだくじなどには感心させられました。特にCrashの名前の由来がわかったときには思わず「おおっ，確かにCrashだっ！」と叫んでしまいました。皆さんもどうかこの感動を味わってみてくださいね。

左の星が華麗に舞います

続いて後半50ステージ(周東さん曰く「〝動〟の50ステージ」だそうです)にいってみましょう。後半の50ステージはそれほど難しいというわけではなく，腕力だけで解けてしまうステージが多いのが特徴です(逆に腕力がないと解けないという説もありますが)。不動ブロックがあまり使用されておらず，ほとんどLRブロックの海なので気長に&てきとーにいじっていれば解けると思います。といっても私はやはり10ステージほど解けずに残ってしまいましたが。後半50ステージに関しては最小ステップが提示されていなかったので，最小ステップ破りの楽しみがなかったのがちょっと残念でした。確かにこういった種のステージで最小ステップを算定するのはかなり骨の折れる作業だと思いますけどね。

謎の13ステージの解法は？

続いては益山さんの作品です。全13ステージなので「今度こそ全ステージ制覇を！」と意気込んだのですが，さすがに数が少ないだけあって絞り込まれていました。周東さんの分類を拝借しますと，主にマイマイステージ，ノーマルステージが多いのです

これが感動のCrashステージ

シューメーカーレビー彗星。う〜ん時事ネタ

ちゃいました。さすがに周東さん，益山さんの作品をこなして私の頭も少しは「PUSH BON!」してきたようです。じーっと画面を眺めているうちに解法が浮かんできました。

しかし最小ステップ（お手紙によると14ステップ）より3ステップ多い17ステップでしか解けなかったので，真の意味で解いたことにはならないのですが。うーん，おかしいなあ，1ステップの無駄もなく解いてるつもりなんですけど。どーやっても17ステップ以下にならない。うがが一っ！ 右端のスターブロックを運んでくるという基本的なアプローチは間違ってないと思うのですが。もしまったく違うアプローチでないと14ステップにならないのであれば，完全に私の負けでございます（これしかアプローチが思い浮かばないんで）。うーん，あと3ステップか。

疲れた脳に一服の清涼剤

続きまして河内さんの作品です。私はとうとうやりました。やっと，やっと達成できたのです。え，なにをですって？もちろん待望の全ステージ制覇でございますよ(作者別のですけどね)。あたしゃ嬉しくて思わず小躍りしてしまいました。ふんふんふふん♪

河内さんの作品は簡単，というよりとても素直なものばかり。そうですね，素性がよいとでも表現したらいいのでしょうか。そこそこ難しいんですけど，変にひねくれた仕掛けがないのでクリア後になんか「また人間を信じられる気持ち」にさせてくれます（ほんとーに難しいステージばっかりやってると結構人間不信の疑心暗鬼に陥ってしまったりするんで）。

スマートな正統派

さてお次は加藤さんの全40ステージを堪能させていただきます。お手紙によると「私が解けるステージしかないので必然的に難易度は低くなっていると思います」とのことですが，まったくそんなことはありませんよ。私もまだ解けていないステージが16ステージほどありますんで（またそれか)。前出の4名の皆さんに比べても決して劣らないレベルです。

全体として左右対称なステージ，スターブロックが隅に配置してあるステージが多いような印象を受けました。19ステージで最初のノーマルブロックを右に押すとちょっと変わった現象が見られますが，これはバグなのでしょうか？ これを利用したステージを作っても面白いかもしれませんね（でも実際そんなステージがあったら絶対に解けないと思うけど)。

ここでふと思ったのですが，周東さんがいうところのあみだくじステージやWaveステージみたいなカラクリステージっていうのは，一度は作ってみたくなるものなのでしょうか？ いえ，加藤さんの40ステージの中にもあみだくじステージが2つほどあったので。確かにLRブロックの反射でブロックが自動的に動いていくさまは，見ていてぐっとくるものがありますもんね（美しいというか)。難易度抜きにしてそういう動きの美しさだけ見せてくれるステージというのも面白いかもしれませんね。

が，1ステージだけどうしても解けなかったので，お手紙にあった答えを見てしまいました。ああ，私の頭脳ってとっても貧弱なのね。

で問題の解けなかった13ステージ目なのですが，益山さんの解法を見てひと言。「こりゃ解けんわっ！（いやちゃんと解けてるんですけど)」。あれだけ複雑怪奇そうに見えるステージがたった5ステップで解けてしまうとは！ 最後の星の動きのスマートさにはちょっと感動すら覚えてしまいました。脱帽ものです。

地獄のステップ1,2,3！

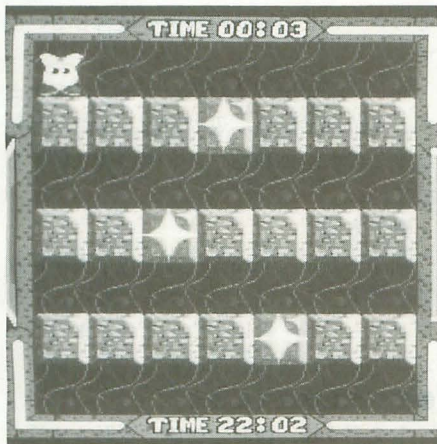
さてお次は一球入魂の三沢さんの作品です。三沢さんの応募ステージ数はなんと1ステージだけ！ お手紙にも「我ながらなかなかよくできたステージだと思います」とあり，難ステージへの自信をうかがわせます。そして，十分後。すいません，解け

どうしても17ステップの壁が……

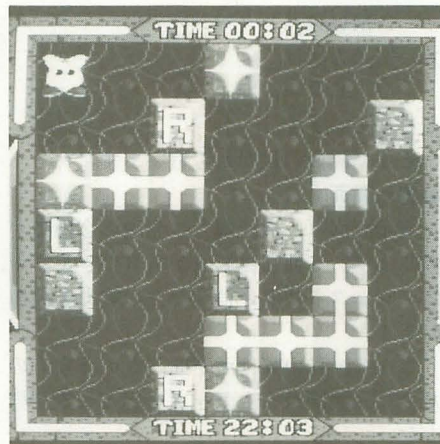
素直さが光ってます

なぜか石の中にいる……？

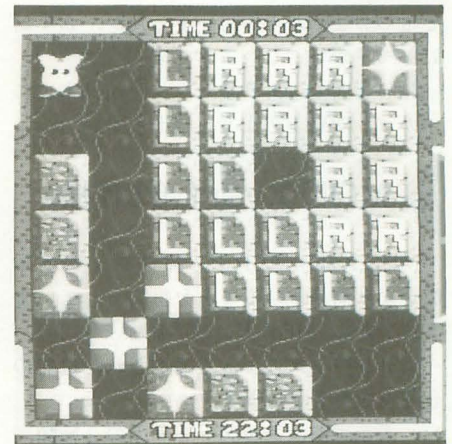
超難問の50連チャン

さて，最後に高橋さんからいただきました全50ステージにチャレンジしたいと思います。思いますが……。実は締め切りぎりぎりまでねばったんですが，まだ12ステージしか解けてないんです。いや，決して遊んでいたわけじゃないんです，信じてください。とにかく高橋さんの作品は難しいんですよ。解けた12ステージもほとんどがサービスステージみたいなもんなんで，実質的に高橋さんには全面降伏してるようなものなんです。うーん，これを全ステージクリアできる人はそうとうな達人だと思います。皆さんもぜひこの辛さを味わってみてください。

脳味噌溶けそー

果てました。やはり210ステージも「PUSH BON!」すると知恵熱も出るってもんです。途中何度か熱暴走しそうになって，脳味噌につけるヒートシンクはないもんかと秋葉原中を探しまわったほどですから(とかいってみたりして)。もうスターブロックは必ず隅のほうに，しかもバラバラに置いてあるし，LRブロックは異常増殖してるし，不動ブロックは必ず置いてほしくない場所に置いてあるし……。スターブロックが6つも7つもあるステージ（これは反則ギリギリ）を見てると，ほんと基本が懐かしくなってしまいます。

それにしても人によっていろんな思考方法があるんだなあ，ということを実感させられました。それぞれの作品にそれぞれの個性が色濃く出てるんですよね，これが。何ステージか解くとだんだんクセがわかってきて，あとのステージが楽になったりして。あと790ステージくらい解いたら，ステージを見ただけでその人の性格とかがわかるようになってしまうかもしれませんね。

ま，そんな冗談はさておいて，皆さんもぜひ今回のもみじ狩りPRO-68Kに収録されている投稿オリジナルステージを遊んでみてください。秋の夜長がすべて「PUSH BON!」一色に染まってしまうことでしょう。そして素晴らしいアイデアがひらめいたら，オリジナルステージを作ってみてくださいね。投稿お待ちしております。私は誰の挑戦でも受けるっ！……たぶん。

いきなり頭を抱えた1ステージ目

がんばれば解けそうにも見えるが……

お手上げです

「PUSH BON!」オリジナルステージ大賞は?

それでは，投稿されたステージの中から，優れたステージを表彰する各部門賞の発表を行います。と思ったのですが，どのステージもそれぞれの味があり，どれも甲乙つけがたい出来。選考する側としてちょっとばかり困ってしまいました。いろいろ悩んだ末，結局以下のように決定しました。

それでは，各賞を発表していきます。

●「なあんてパワフルな人なんで賞」

周東　正男

ひとりで100ステージものオリジナルステージを作成してしまったすさまじいパワー。そのパワーに敬意を表して，周東さんには「なあんてパワフルな人なんで賞」を進呈させていただきます。

●「スターブロックが華麗に舞うで賞」

益山　直人

なんといっても，13ステージのスターブロックの動き。目を見張るくらいの美しい解法に惚れました。

●「山椒は小粒でもピリリと辛いで賞」

三沢　弘之

小粒ながらもやっぱりヒトクセある1ステージを投稿してくださった三沢さん。これからもいろいろ挑戦してみてね，という思いを込めて。

●「スカッと爽やか・脳の清涼剤賞」

河内　一真

もう，脳味噌がでんぐり返ししてしまいそうななかで，さっくり全ステージクリアできた喜びを与えてくれた河内さん。ありがとう。

●「シンプル　イズ　ベスト賞」

加藤　和人

オーソドックスで堅実な作りながらも，確実に私の脳味噌をいじめてくれた加藤さんのオリジナルステージ。どのステージもおいしかったですよ。全40ステージをクリアできなかった悔しさも一緒に進呈しちゃおうかな。

●「超絶技巧難関賞」

高橋　貴之

とにかく高橋さんのステージは仕掛けの発想が違うので，いままでやってきたノウハウがことごとく通用しなかったりするんです。これが痛い。1ステージ目から30分くらい「これは絶対解けないんじゃないの？」と頭抱えてましたから(なんとか1ステージは解けましたが)。いやー，本当にまいりました。

ということで，各賞の発表を終わります。次はいよいよ大賞の発表です。ドルゥゥゥゥ（ドラムの音ね）。

●「PUSH BON!オリジナルステージ大賞」

高橋　貴之

パンパカパーン！　高橋さん，おめでとうございます。本当にそれぞれのステージが，とてもクオリティが高く，ひたすら私の脳味噌をいじめてくれました。その努力とセンスを評価して「PUSH BON!オリジナルステージ大賞」をあげちゃいます。え～い，もってけドロボウ！

発表を終えて

「なあんだ，結局，全員がなにかしら受賞するなんて審査が甘すぎるんじゃない」，なんて言葉には屈しません。だって，本当に面白かったんだもの。読者の皆さんもちゃっちゃか付録ディスクから解凍して，実際に遊んでみてください。また，「PUSH BON!」を新しい気持ちで遊ぶことができますよ。

なお，今回，ディスクに収録された方々には，ディスクに収録された基本原稿料のほかに「ふぇふぇぅぇぅぇ」を差し上げます。そして，大賞の高橋さんにはお好きなゲームソフトを1本プレゼントしちゃいます。

楽しみに待っていてください。

粘土細工のようにモデリングができる?

MOD.X ver.2.0

Tsuboi Hiroshi 坪井 浩

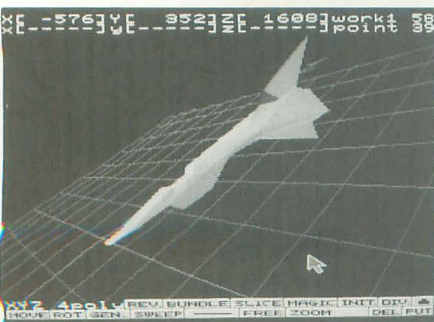
機能が大幅アップしてさらに使いやすくなったSLASH用新モデラ
今度こそ，粘土細工のようにポリゴンを扱えるようになりました
この言葉がハッタリかどうかはあなた自身の手で確認してみてください

前回レポートしたモデラを，使えるモデラに仕上げてみました。概要は，モードレス＋スライサー＋３面図＋……と，詰め込んだ感じとなっています。メッシュ変形などもつけようかとも思いましたが，さすがにメモリが足りないようです。

はったりではありません。今回は本当にポリゴンを粘土のように扱えます。

MOD.Xver.2.0の仕様

MOD.XはSLASH上のオブジェクト生成をより簡易化することを目的としています。まずは，MOD.Xの基本仕様，今回のバージョンアップによって追加された機能の仕様を紹介していきます。

1) 表示画面は透視図と３面図の２種類
2) 自動面破綻回避，スライス，回転生成機能の内蔵
3) 遅いマシンに捧げるバウンディングボックス表示により速い応答の実現
4) SLASH ver.3.00（暫定版）を利用して表示
5) 編集用（作業用）バッファ，アンドゥバッファ，最大256個のポリゴンが登録できる編集用ワークが３つ，さらにそれぞれのワーク間で合成が可能
6) ファイル入出力は，拡張されたPLG方式（ベタカラーに対応）
7) 要2Mバイト

透視図

使用方法

起動方法は，コマンドラインより，

＞MOD〈リターン〉

で，終了方法は，

Q〈リターン〉

となっています。

まずは画面の説明から行います。

●表示画面

表示画面は，透視図と３面図の２種類があります。３面図は表示がちらつきますが，オブジェクト作成にはかなりの威力を発揮します。

さらに，私の脳は亜空間，という貴方のために３面図からもデータを起こせる仕様となっています。

・物体座標系の中心座標と指点座標

画面左の一番上にある数値群は，現在表示されているポリゴンの物体座標の中心が，SLASHの絶対座標系内のどの位置であるかを示しています。

そのすぐ下の数値群は，マウスで指した点（指点）または３面図での十字カーソルが示す点が，物体座標系内のどの位置であるかを示しています。

・workN/ポリゴン数

Nが現在作業中のワーク番号を示しています。その右隣の数値はそのワークのポリゴン数を示しています。pointの右隣の数値はそのワークの頂点数を示しています。

・XYZ/単位移動量

画面下方にXYZシンボルが表示されています。これは，コマンドの実行に際してそのコマンドの補助的制御を担当しています。

同じく画面下方に数値が表示されています。単位移動量は，辺の伸縮，頂点の移動に際しての単位移動ベクトルの長さのことです。

・表示方法

画面下方に，wireまたはpolyと表示されています。これは，ポリゴンの表示方法をワイヤーフレームにするかそのままのポリゴンで行うかを示しています。

●モデラを構成する要素

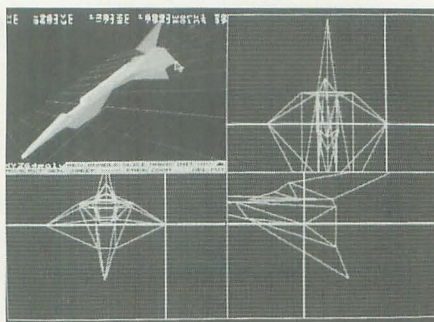
本モデラは，雛形の物体を変形させて，部品化し，最後に合成するというスタイルで使用すると最も威力を発揮します。ここでは特に，変形の際にユーザーが理解しておきたい事項を説明します。

・点

点には，指点，固定点，自由点の３種類があります。

指点はマウスカーソルでポリゴンの頂点を指すと黄点で示されます。

ユーザーがモデラに登録できる点は，固定点と自由点です。固定点は文字どおり頂点が固定である属性をもっています。一方，自由点は，以下に示す頂点の移動などに際して，自由に移動できる属性をもっています。

両点の登録方法について説明します。透視図で頂点を登録する方法は，マウスカーソルで登録したい頂点を指点で指し，マウスの左ボタンで固定点はダブルクリック，自由点はシングルクリックします。両点の解放は，解放したい頂点を指点で指し，マウスの右ボタンをクリックすることで行えます。ただし，解放はマウスカーソルが透視図内にいるときのみ有効です。３面図よ

３面図

り頂点を登録する方法は，マウスカーソル移動と右クリックで十字カーソル（白線で示されている）の位置を指定し，先と同様の左クリックで行えます。

画面では，固定点は白色，自由点は水色の点で示されます。

前回のモデラはクリック処理の不備でかなりの問題がありましたが（致命的ともいう），今回は大丈夫です。

・辺

辺は固定点，自由点いずれかの２点より構成されます。登録できる辺は編集中のオブジェクトの頂点間です。

始点と終点を先の方法でモデラに登録すると，登録した点より構成される辺が登録されます。辺(縁)は連続して登録できます。辺の解放方法は辺を構成する任意の頂点を右クリックすることで行えます(マウスカーソルが透視図内にいるときのみ有効)。

画面では，ポリゴンの辺(縁)である辺は白色の線分で，そうでない辺は赤灰色の線分で示されます。

なお，このモデラにおいては，ポリゴンを辺で囲むことが頻繁に行われると思います。よって，入力された辺によってはモデラが勝手にポリゴンを辺で囲ってしまいます。注意して登録してください。

・面

本モデラで扱えるポリゴンは三角形と四角形のみです。四角形は頂点の移動などで面に破綻が生じるときがありますが，モデラが勝手に三角形分割してくれます。

●ユーザーインタフェイスと機能説明

画面を見るとスイッチが配置されています。スイッチはマウスカーソルで指し，マウスクリック＋移動などで使用できます。スイッチを押しているときは，スイッチの表示が青っぽくなっています。

左ボタンがプラス，右ボタンがマイナスという感じで使用してください。

キーボードから利用する方法は相変わらずワンキーメニュー方式です。メニュー中では，カーソルキーでメニュー選択，リターンキーで実行，ESCキーでメニューからの脱出です。

さて，各機能の説明を行います。機能に対応するキーボードのキー，[]で囲まれたものがスイッチのシンボルに対応し，そして機能の説明という形式で行います。

・Q

モデラを終了します。

・カーソルキーおよび**OPT.1，2**

表示する物体座標系の角度を変化させます。画面上方の大文字のXYZシンボルをクリックして変化させることもできます。

・テンキーおよび**XF1,2**

物体座標系の位置を変化させます（視点の変化)。テンキー4,6,2,8はXY座標，XF1,2キーはZ座標を変化させます。３面図では，テンキーの4,6,2,8が３面図の表示中心のXY座標の変化を，５がZ座標の変化を受けもっています。先と同様に画面中の適当な数字をクリックすると物体座標系の位置が変化します。

・XF5

表示方法を切り替えます。画面中の「wire」もしくは「poly」と表示されている部分をクリックすることでも行えます。

・A

ZX平面の表示を制御します。

・HOME

視点を初期状態に戻します。

・TAB

透視図＋３面図の表示方法の切り替えを行います。

・CTRL,SHIFT

３面図の表示の拡大縮小を行います。

・O

雛形の物体を生成します。

・C

ポリゴンの色情報を変更します。辺で囲まれたすべてのポリゴンの色を変更できます。またOキー，回転生成で生成する物体の色も指定できます（その場合はベタ塗りは無効)。

・CLR

作業用バッファを初期化します。

・XF3,4　[　──　]

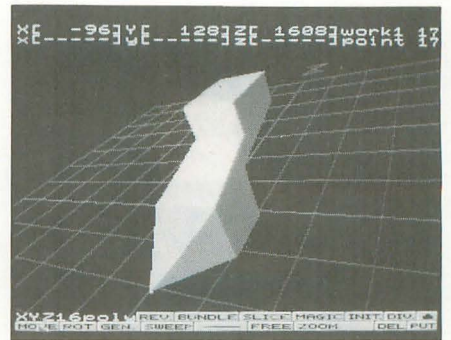
辺の伸縮を行います。XF3キーまたは左ボタンで辺が伸び，XF4キーまたは右ボタンで辺が縮みます。伸縮の方向ベクトルは，辺が登録されたときの辺の単位ベクトルに固定されることに注意してください。

ただし，その移動量は，画面中の下方に示されている数字をクリックすることで変更できます。

・[FREE]

登録されているすべての辺を解放します。また，編集バッファの最適化も行います。ほぼ重複している頂点（半径３ドット内の頂点）の排除や，ポリゴンの形状の最適化などを行います。

・スペース　[SWEEP]

辺に囲まれたポリゴンの引き出しを行います。SWEEPを行ったあとに，辺の伸長（[　──　]ボタンの左クリックなど）で，それぞれの法線方向に同形状のポリゴンが引き出せます。その逆も行えます。

自由自在に形状作成

・D　[DIV.]

辺で囲まれたポリゴンを重心で三角形に分割します。

・P　[PUT]

辺で囲まれたポリゴンに，それぞれ同形状の大きさ1/4のポリゴンを重心を共通にして生成します。

・DEL　[CUT]

辺で囲まれたすべてのポリゴンを削除します。

・BS　[REV.]

辺で囲まれたすべてのポリゴンの頂点並びを逆順にします。

・G　[GEN.]

各辺をモデラに登録する際に最初に登録した頂点を第１登録点と呼ぶことにします。そしてモデラに３辺だけ登録されているなら，その３辺の第１登録点からなる三角形を生成します。４辺以上ならモデラが最初に発見した４辺の第１登録点よりなる四角形を生成します。

また，同形状（ここでは，頂点と法線方向も同じの意）のポリゴンが存在した場合はポリゴンを生成しません。

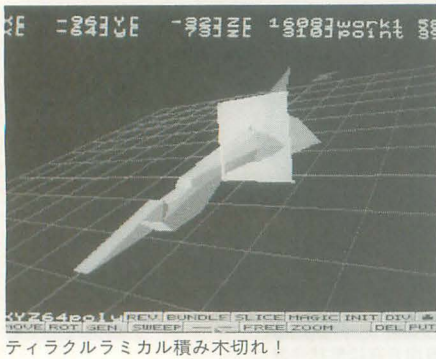
左，右ボタンで交互に[GEN.]ボタンを押すと両面にポリゴンを生成します。

・[MOVE]

自由点を移動させることができます。左ボタン＋マウスの横方向の移動で物体座標系でのX方向の移動，左ボタン＋マウスの縦方向の移動でZ方向の移動，右ボタン＋マウスの縦方向の移動でY方向の移動が行えます。

なお，各軸の移動の制御は画面下方のXYZシンボルをクリックすることで行えます。シンボルが反転しているとき，その軸の移動は不可であることを示しています。

・[ROT]

モデラに登録されている自由点の重心を回転の中心として，物体座標系内で自由点を回転させます。MOVEと同様に左ボタン＋マウスの横縦移動がXZ軸回転，右ボタン＋マウスの縦移動がY軸回転となって

ティラクルラミカル積み木切れ！

います。画面に表示されるパラメータはSLASHver.1.00でのパラメータです。軸の回転の制御は先と同様です。

・＿　[MAGIC]

ポリゴンをスライスするための板の表示を制御します。板（魔法板）が表示されているときは，[MOVE][ROT]とも魔法板にのみ作用することに注意してください。

・／　[SLICE]

魔法板（領域のある平面）でポリゴンを切ります。断面はワイヤー表示のとき，青の線分（板の辺の色と同じ）で示されています。スライスされた片方のポリゴン群は自由点で構成される辺で囲まれています。いったん魔法板を消してから，MOVEで分離することもできます。

・．　[INIT.]

板を初期化します。

・[ZOOM]

拡大/縮小を行います。対象となる物体を辺で囲み，左ボタンで拡大，右ボタンで縮小します。画面下方のXYZシンボルにより拡大の仕方を指示できます。反転しているシンボルがその軸の拡縮を行わないことを指示します(例：Y方向だけの拡縮→XZシンボル反転)。

・[凹/凸]

頂点の移動により面が破綻したときの回避方法を示しています。実際に確認してみれば，どんな動作かがわかると思います。

・S

現在のワークのポリゴン情報を保存します。ファイルフォーマットはPLG形式です。直接ファイル名を指定するときは，任意のディレクトリに移動してから（パス名は結合される），ファイルメニュー中でUNDOキーを押して入力してください。拡張子は自動的にPLGになります。

なお，ドライブの変更はカーソルキーで行えます。

・L

現在のワークにポリゴン情報を復元します。ファイルフォーマットは先と同様のPLG形式です（以前のPLG形式のファイルも，もちろん使用できます）。

ロードを実行すると作業ワークの以前の内容は失われます。

・M

ワークを合成します。

・I

辺で囲まれたポリゴンの頂点を分離させます。収録後，バグが発見されました。使わないでください。

・B　[BUNDLE]

ポリゴンの辺はグラフとみなせます。そして，現在登録してある辺が属する木を構成する辺すべてを登録します。

・E

拡張機能を呼び出します。ポリゴンソート，対称生成，反転，回転生成という機能が利用できます。いずれの機能も作業対象のワークに作用します。

ポリゴンソートは，ポリゴンの優先順位を自動的に決めて並べ替えることを行います。ポリゴンソートの詳細は，本誌1993年11月号の丹氏の記事を参照してください。

対称生成は，ポリゴンを各平面に対称にして重複しないように生成することを行います。

反転は，各平面に対してポリゴンの位置を反転させます。ポリゴン数は変化しません。

回転生成は，伸縮辺より等分割の回転体を生成します。分割数は適当に指定してください。表示がおかしいと思ったら[REV.]コマンドなどで，頂点並びを逆順にしてみてください。

・X

辺で囲まれたポリゴンをワークに複写します。

・ROLL UP/DOWN

辺で囲まれたポリゴンの順位を手動で変更します。ただし，辺で囲まれたポリゴンは1つだけにしてください。ROLL UPで優先順位を1つ上げます。ROLL DOWNで優先順位を1つ下げます。

・F1/2/3

全部で3つある作業ワークの1つを作業の対象に変更します。FNのNが作業対象とするワークの番号に相当し，その番号のワークに作業ワークを切り替えます。

・F6/7/8

ワークの表示を制御します。6がワーク番号の1に相当します。作業の対象でないポリゴンは青黒い線分で囲まれています。3面図では青の線分で表示されています。

・UNDO

アンドゥバッファの内容を作業用バッファに転送します。

・H

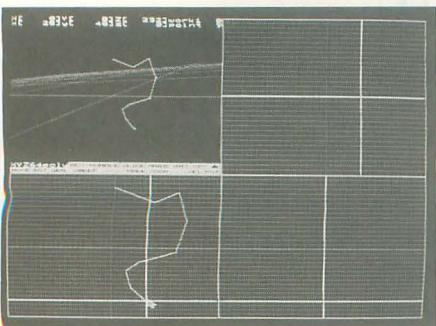
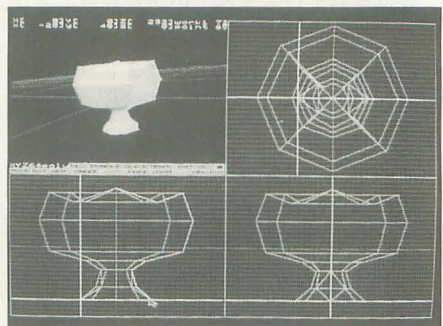
ヘルプメニューを表示します。

動作確認

それでは実際に使ってみましょう。起動すると，画面には数値とボタンとマウスカーソルだけが表示されています。まず，カーソルキー↓でほとんど意味のないZX平面を表示してみましょう。

次に，Oキーで物体を生成してみましょう。立方体を選んでください。XF1,2またはカーソルキーで物体を見やすい位置にして，マウスカーソルで透視図内のオブジェクトの頂点を指すと，黄点で示されている指点が登場しました。

次にマウスの左クリックで頂点を登録してみましょう。頂点を次々に指して左ボタンを押すと白または赤灰色の線で示される辺が現れたと思います。あるポリゴンを囲むように連続して登録していくと，ポリゴンが勝手に囲まれてしまうことも確認してみてください。

基本は辺の伸縮です。辺を登録したなら，[　―｜]ボタンをマウスカーソルで指してボタンを押してみると辺が伸びたり縮んだりします。ポリゴンの辺ではない辺(赤灰色の線分で表示されます）も登録して伸縮をさせてみてください。山吹色の線分による三角形がある面を覆っているときは，

この伸縮辺より回転生成を行うと

このように生成します

その面が破綻していることを示しています。何度もいうようですが，面の破綻はモデラが勝手に回避します（これを見たら，ポリゴン＝粘土と思うかも)。回避方法の指示は[凹/凸]ボタンで２通りできることの確認もしてみてください。

次は，SWEEP機能を試してみましょう。ポリゴンを辺で囲って，[SWEEP]ボタン，そして[　──　]ボタンを押してください。面が引きずり出されています。

画面下方XYZの横に示されている数値をマウスカーソルで指し，クリックで変化させてください。そして，[MOVE]ボタンを押しながらマウスを移動させてみましょう。画面中のオブジェクトが動いていますね。[ROT][ZOOM]もついでに試してみてください。と，こんな感じでひととおりの機能を試してみましょう。

効果的な利用方法

各機能の効果的な利用方法について記してみます。

まず，ポリゴンの辺ではない辺ですが，ある物体を対称的に変形させたいときに使えます。３面図を物体の移動，合成のときに参照することでより感覚的に行えると思います。

自動面破綻回避とスライスそして複写機能により，目的の形状が楽に作れるようになったと思います。この機能をさらに生かすには，作成したい物体の外観を紙面などにスケッチして徹底した部品化を行って，部品作成，合成，部品作成，合成，という利用方法が最も効果的だと思います。

また，表示の速度を考慮した場合，なるべくポリゴン数を少なくするように心掛けることはいうまでもありません。

備考

MOVEなどの処理が重たくなってくると，対象となっている頂点を緑のバウンディングボックス（境界箱）で表示します。遅いマシンでも，それなりに使えるように配慮してみました。

ポリゴン情報の最適化は[FREE]ボタンを押すと行えます。

ポリゴンをスライスして，スライスされた平面が五角形になってしまったときは，そのポリゴンは三角形と四角形に自動的に分割されます。なお，分割のされ方が気にくわないときは，手動でポリゴンを分割し直してください。表示方向がある程度一定のときは，縦ループをなくす方向で分割すると表示が若干高速になります。

魔法板を表示しているときには，MOVE/ROTともに辺に登録した頂点の属性を無視します（辺の伸縮は別)。

スライスの動作ですが，若干の手抜きにより，誤差と仕様の不備があります（たとえば，同じところで何回もスライスすると，交点の状態によってはポリゴン数がどんどん増えていくなど)。

しかし，注意して利用すれば，実用には耐えられると思うので問題はないと思います。

SLASHの表示原理を理解していない人が陥りやすい事項としては，GEN.コマンドの動作があります。「正しくGEN.したのにpolyでみたら，そこにはなにも表示されていない。バグか？」答えは，NO。SLASHはポリゴンの法線と視線の内積値で可視，不可視を決定しているからです。よって，なにも表示されていなかったらば，REV.コマンドで頂点の並びを逆順にしてください（法線が逆方向を向きます)。

また，作業用バッファ→アンドゥバッファの転送は指点がマウスで指せる領域でのマウスの右クリックのたびになされます。

MOD.Xの作成方法に関してですが，使用するライブラリはXCver.2.1の旧バージョンのものを利用しています(XCのライブラリならばたぶん大丈夫)。ただし，LIBCのものでは動作しないと思います。

コンパイルに際しては，各自でmod.h/makefile/slashlinkを書き換えてください(ちなみにコンパイルには20〜30分くらいかかります)。前回はこの点に関して迷惑をかけました。

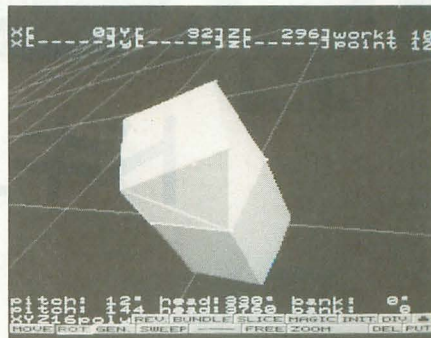
データ利用法

本バージョンより，ユーザーの作成した色情報がモデラ内に登録できるようになりました。登録の方法はslcol.idxというテキストファイルを介して行います。

まず，環境変数SLCOLにslcol.idx（小文字のファイル名のみ有効）へのパス名を登録します。

slcol.idxのフォーマットは次のようになっています。

```
フラットカラーファイル名
        ⋮
------------
ベタ塗り色名　カラーコード
        ⋮
```

フラットカラーファイルはバイナリファイルで，モデラは先頭の128バイトを取り込みます。モデラ内での色名は主ファイル名になります。ベタ塗りのカラーコードは，４バイトの16進数です。

面の破綻を気にする必要はない

ファイルの単語の区切りは１つ以上の空白かリターンコードとなっています。

フラットとベタ塗りの区切りは１つ以上の連結された記号群“-”となっています。また色名は14文字以内にしてください。

●slcol.idxの使用例

```
・AUTOEXEC.BAT
    ⋮
  SET SLCOL=D:¥v3_slash¥col
    ⋮
  [EOF]
・slcol.idx
  COL1.PAL
  COL2.PAL
----------------
  betacol1 ffffeeee
  betacol2 aaaa1111
  [EOF]
```

PLGフォーマットは拡張されましたが，従来のフォーマットによるデータはそのまま利用できます。また，PLGフォーマットの拡張についての詳細はPLGCONV.Cを参照してください。

今後の展望

現時点では，モデリングにおける座標系は１つしかありません。単純な物体ならばそれで十分なのですが，たとえば，人体のデータを作成するときには座標系が重要となりそうなことは明白です。

やはり，構造体できちんと考える必要がありそうです。野望実現のためには，焦らずに，１つひとつギャップを埋めるしかありませんね。

そういうことで，(突然人格が変わる）３D野郎ども，自身・他人のパワーをフルに活用して自らの野望に突き進め(BGMはブルックナー，９−１)。期待している。

TeX入門講座 〜てふてふらてふ〜

ドキュメント作成の巻

Taki Yasushi 瀧 康史

「X680x0 TeX」はもう購入されたでしょうか?
買ったはいいけれど，どうやって使えばいいのかわからない
そんな人は，簡単なところからいろいろ試していきましょう

綺麗に文書整形しましょう

「てふてふらてふ，らてふてふ」。なんて歌はない。なんとなくいま作った。

新しく始まったTeX入門である。

身の回りにTeXを使いたいけど，よくわかんなーいという人が，あまりにもたくさんいるので急遽，筆をとることにした。私自身，まだまだTeXnician＊1には程遠いが，実験レポートや，論文，フリーソフトウェアのドキュメントぐらいはTeXで書いている。個人的にはTeXで論文を書くのは，ZMSファイルで音楽を作るよりずっと楽だと思っている。

＊1 (てふにしゃん)どうやらTeXのテクニシャンのことをこう呼ぶらしい。

この記事は誰のため?

この記事は，いままさにTeXをやろうと思っている人，そしてTeXに憧れを抱いていて，最近発刊された「X68k Programming Series X680x0 TeX」以下「X680x0 TeX」)を購入したばかりの人，こういう人をターゲットにする。

いままでワープロを使っていたが，TeXというシステムに興味がある。

これも，よい。要は興味を持っていること。

それでは環境について話をしよう。この記事は「X680x0 TeX」を買って普通にインストールした状態をベースにしている。

TeXをアプリケーションとして使う立場から話を進めるので，私のほうからTeXのインストールとか，システムについて込み入った話をするつもりはない……というかできない。TeXを使い始めてからはわりと長いけれど，いまだにTeXに関していえば，私はアプリケーションユーザーだからだ。

どのように進めるか

記事の進行について話しておこう。

まず最初に印刷結果とソースを比較して載せる。ソースは打ち込めといっているわけではない。参考にしてほしいだけ。とりあえず人のTeXファイルを見て真似をする。私自身そうだった。自分自身オリジナルのTeXファイルを作るのは，TeXに興味を持ち，マニュアルを自分から進んで見るようになってから。それまでは，スケルトンでも用意して毎回それをいじればいいのだ。

したがって，本文よりもひょっとしたら，TeXのソースや印刷結果のほうが多いかもしれない。基本的には，私自身が初めてTeXを勉強した際，知りたかったことを何回かに分けて掲載するつもりである。

コマンド一覧などを書くつもりは毛頭ないので，知りたい方は「X680x0 TeX」の資料や，その類の本を読んでほしい。

さて今回は記念すべき第1回。

まずはTeXに慣れていただこうというわけで，今月の目標は，数式の出ない，フリーソフトウェアのドキュメントをTeX形式にすることだ。

ちょっとした規則

ソースの解説をしていく前に，TeXに関する常識というものをまとめてみたいと思う。

まず，TeXは自動で組版するので，エディタで変な空行などをあけても意味はない。裏技を使うとすれば，全角スペースなどもあるのだが，これを使うのは当然美しくない。

それから，英文字，数字は，基本的に半角で打つ。半角で打てばTeXは自動的に全角と半角の間にちょうどよいスペースを入れるなどして，見かけをよくしてくれる。私は半角をよく使うせいで，日本語FEPをCTRL+XF1ではなくXF5だけで起動するようにした。XF5はいままで確定ぐらいにしか使っていなかったので，あまり重要ではなかったからだ。日本語FEPの起動方法の変更の仕方がわからない人は「日本語入力・辞書ユーティリティユーザーズマニュアル」をちゃんと見るように。

これで速いHDDかRAMDISKに辞書を載せると，XF5を押すことにより，半角モードと全角モードが瞬時に切り換わる。TeXは英文字と全角文字を混ぜ合わせることが多いので，できる限り素早く切り換えられたほうが都合がよいからだ。

あとは必要になったら，その都度覚えればよい。あぁ，実に簡単。

サンプル

まずは印刷結果とリストを，交互に眺めてほしい。これは私がフリーソフトウェアのドキュメントを作成するときに使用しているスケルトンだ。

いまからこれらの機能について説明する。もしもあなたが作りかけのソフトのドキュメントをTeXファイルでやろうと考えているならば，私と同じフォーマットでやれば，ハンドルなどを書き直すだけで話はす

む。

それでは順に読んでいくことにしよう。

1ページ目

たいそうな目次がついているが，これは私自身がチマチマと作成したわけではなく，TeXが自動作成したものだ(図1)。大半のワープロソフトに比べて，こういうことを自動で作成してくれるというのがTeXの嬉しいところか。

まず，リストと対比してそれぞれの意味を説明していこう。今後特に断りのない限り，n行目という表記はリストのn行目をさすことにする。

1行目。

¥documentstyleは，TeXに組版するスタイルを教える命令。これは最初に必ず書いておく。

次の［a4j］だが，これは印刷したときの用紙サイズを意味する。たとえばB5にしたければ，［b5j］とすればよい。{jarticle}はこういうフォーマットのタイプで印刷せよという指定。これについてはいまはいろいろいうつもりはない。

3行目のtitleはいうまでもないだろう。4行目についても作者名を記述するのみ。5行目の¥dateは日付を入れている。私はTeXソースを書いた日を入れるために，¥todayというその日付を返すマクロを利用している。

実際の印刷に相当するものは，7行目から始まる。最初の¥begin{document}は，最後の378行の¥end{document}にネストしていて，このネストのなかに実際の文章が入ることを意味している。

8行目の¥maketitleは，3，4，5行で定めた条件にしたがって，タイトルを書くことを意味している。いちいちスペースをあけたり，センタリングなどをしなくても，タイトルはTeXが作ってくれるのだ。

9行目は目次を自動作成してくれるマクロで，10行目は，表目次を作るマクロ。11行目はなくてもよかったかな。13行目は改ページを意味する。

タイトル＆目次ページはこれでできあがりだ。

2ページ目

16行目の¥section命令は，基本的に「節」を意味している。これは1ページ目(図1)を見てわかるとおり，目次に反映される。

段落は基本的に，空行で表しているので，適当なところでリターンすればよい。本誌は，リストを基本的に64桁で改行するため，今回のリストもできるだけそれに合わせた。その結果，いまの位置で改行しているわけであり，途中で空行リターンや¥¥が入らない限り，それは同じ段落を意味する。

23行目の¥LaTeXは印刷結果どおり(図2)，あのかっこいいLATEXのロゴを表示する命令である。このような¥から始まる命令は，命令後に半角スペースを入れることを忘れずに。今回のソースではわかりやすくするために¥LaTeXの前後に半角スペースを入れておいた。

このようなロゴはほかにもいろいろあるので，調べてみると面白いかもしれない。

個人的には，X680x0ロゴとか誰か作ってくれないかなと思っているんだけど。

26行目の¥section{使用許諾}で節が変化する(図3)。文章はただ単にツラツラと

図1　目次

PCMPLAY Ver.1.21

蘶焔手紅呪威羅黄爵神

平成6年 8月 22日

目次

表目次

図2　LATEXのロゴに注目

1 概要

このドキュメントは限定ドキュメントです。
連載のために、取捨選択しながら書き上げてい
本原稿は LaTeX システムを利用することにより

図3　節が変わった。最後の行の「アーカイブ[1]」に注目

2 使用許諾

pcmplay.x は無料です。今後、有料になる可能性はまったくありません。よって自由に使用し他人に譲ることもできます。ネットワークや、BBS にアップロード、商的目的で使用することに関しては全く自由です。

ただし、pcmplay は全くのフリーソフトウェアではありません。pcmplay は著作権法で保護されていますし、配布に関しては制限もあります。しかし、この制限は、善意の使用者の行動を禁じないように定めることにします。禁じられているのは、第三者への譲与の拒否、すなわち、pcmplay の普及を妨げることです。このような事をする人に、pcmplay を使う権利は有りません。

pcmplay は無償で使用許諾されるので、著作権法ほかの日本国内の法律で許される範囲での保証はいっさいしません。このドキュメントに記載された資料においてもです。そしていかなる場合も、私、蘶焔手紅呪威羅黄爵神及び、pcmplay を配布した第三者は pcmplay とそのアーカイブ[1] を、まったく保証なしに提

書いてあるだけ。英数字は普通に半角で書いてある。このリストは基本的に64桁改行しているが，基本的には句読点など，見やすいところで改行すればよい。

44行目の¥footnote{}は{}のなかを注釈として外に出すことを意味している(図3，4)。印刷結果の対応する部分には，小さな上付き文字1がついているだろう。欄外に小さな文字で，この¥footnote{}のなかに書かれた文字が書かれていることに注目してほしい。

TeXはこのように簡単に欄外に注釈をつけるができるというわけだ。3節目(動作環境)には説明の必要なところは特にないだろう。

3ページ目

このページにはそれなりにTeXらしい使い方が出ている。

まず89行。¥gtだがこれは{}で囲まれたなかをゴシック体にしている(図5)。

ちなみに，TeXでは，英数字を部分的に強調したい場合，{¥em XXXXX}と表記する。これで囲まれた文字列はイタリック(斜体)になり強調される(強調した文字列をさらに強調する場合には，その部分についてはローマンに戻る)が，日本語ではなにも起こらない。

日本語ではある文字を強調したいときに，ゴシック体がよく使われるため，{¥gt これは強調}といったふうに使った。

一般にTeXでは，コマンドや環境に基づくものは，¥*****{ 囲み }で処理していて，同じ条件をもつ団体に関しては，{¥*** 囲み}というように処理する。

実は私はどういうものが環境で，どういうものが条件なのかはよくわからなかったりする。来月までの私自身の宿題にでもしておこう。

94行目，やっとTeXらしい使い方が出てくる。これは，enumerateといわれる環境で，¥begin{enumerate}から，105行目の¥end{enumerate}までネストしている。

この環境は¥item以後の文章を箇条書きにする(図6)。改行はどこで入ってもかまわないし，¥item以後の文章が猛烈に長ければ，うまく段組みしてくれる。まぁ一度試してみればわかるだろう。

122行目，数式環境である。数式環境はいくつか指定があるが，ここでの指定は，文章中に数式を混ぜるために利用する。そのため展開の仕方が小型になるがネストは$から$まで。具体的には，−128(0x80)〜+127(0x7f)が数式モードになる(図7)。英数字はイタリックになるので，いまいち格好はよくないかも。なんのために数式モードにしたかということ“〜”を表示するためである。“〜”は数式記号で，similarといい，近似を表す記号なのだ。「から」ではないから，代用だったりする。この場合には，$¥sim$といったように，¥simだけを数式モードにすればよかったかな?

また，同行の{¥em char}は先に説明した英数字の強調モード。

144行目。

まず{¥small ……で，ネストした}がくるまで，文字を小さくする。この間，別に1行にまとまらなくてはいけないわけではない。文字の大きさは小さい順に，¥tiny，¥scriptsize，¥footnotesize，¥small，¥normalsize，¥large，¥Large，¥LARGE，¥huge，¥Hugeといった具合。詳しいことは「X680x0 TeX」のVol.1，139ページを見てほしい。

145行目は環境verbatim指定。それほど使う環境ではないのだが，リストなどを載せるときに利用する。見てのとおり，入力されたままを出力する環境設定(図8)。

つまり144行目の¥smallと合わせて，本文よりも少し小さめの文字で書き出せという指定をしているわけだ。

図4　図3の「アーカイブ¹」に対応する注釈

> **3　動作環境**
>
> この項では、ソフトの動作環境とインストールの方法を説明します。
> バイナリファイル、pcmplay.xは特に起動のための外部ファイルなどを必要としません。あなたがそのほかのツールをいれるディレクトリにpcmplay.xをcopyないしは、moveすることにより、インストールは終了します。pcmdrvやopmdrvの類は必要ありません。
>
> ¹この場合のアーカイブとはアーカイブの形式まで制限している訳ではありません。また、アーカイブはオリジナルのまま、提供する必要はありませんが、明らかに害になるものを含ませての配布は良心の上で行わないで下さい。

図5　「再生出来るようにする」がゴシック体

> この項では、ソフトの具体的な使用方法を説明します。
> このソフトは、基本的に各種のPCMデータを**再生出来るようにする**目的で作られています。
> PCMのデータ形式は、非圧縮型、圧縮型を含め、多岐にわたりいろいろありますが、現在のバージョン

図6　箇条書きにする

> では、次の種類しかありません。
>
> 1. MSM6258フォーマットのADPCM。拡張子はPCM。
> 2. 符号付き16bitPCM。拡張子はp16,phl。(ビッグエンディアン)
> 3. 符号付き16bitPCM。拡張子はplh。(リトルエンディアン)
> 4. 符号付き8bitPCM。拡張子はpc8,p8。
> 5. 符号無し8bitPCM。拡張子はpn8。
> 6. 符号bit付き8bitPCM。拡張子はps8。(MEGA CD、FM-TOWNS用)
> 7. 謎の拡張子*.PCEファイル。
>
> 1は普通X680x0に内蔵されているMSM6858Vによるadpcmファイルです。X680x0の世界では、*.pcm

図7　数式環境を使用

> 6番は主に、MEGA-CD、FM-Townsなどで使用されている8bitPCMです。
> $-128(0xff) \sim +127(0x7f)$の範囲です。MSBが符号ビットな為、中心点0が2つあります。(0x00,0x80)
> CDROMユーザーで、PCMファイルを集める習慣がある人なら、これらのpcmデータが直接再生でき、

4ページ目

4ページ目にはたいした指定はない。

あえて説明するとしたら，224行目の¥rightarrowか。文字どおりこれはそのまま(図9)。

5ページ目

5ページ目はそれなりに実がある。

まず，237行目。

¥subsectionというのは節よりも小さな区切り(図10)。日本語でなんと呼ぶかわからない。

245行目。¥bigskipは垂直に少し行をあける命令。同じようなものに，¥smallskipなどもある。詳しくは「X680x0 TeX」を参照してもらおう。

さてTeXでの表の書き方を示しているのが，247行目。凝った表ではなく，この程度の表ならば簡単である(図10)。

まず，表タイトルの環境を与える。これは，そのまま表目次に定義される。247行はそういった意味でのエンギものと考えてよい。248行目の¥captionが表タイトルをつける命令。表番号はあえてつけなくても，table環境のもとで管理されるので考えなくてよい。

249行目の¥begin {center} はセンタリングの指定。¥end {center} まで囲まれた部分をセンタリングする。

250行目がいよいよ表を作るtabular環境。¥begin {tabular} のあとの {cr} は表の中身をどちら寄せにするかを指定しているのだ。

最初のｃは左のRate No.をセンタリングすることを意味し，次のｒはSampling Rateを右寄せにすることを指定している。もしも左寄せにしたければｌと書く。

表の列が2つではなく，3つならば，その分も書かなくてはならない。たとえば3つともセンタリングで書きたいならば，¥begin {tabular} {ccc} のように。

この環境では表を区切る縦線を書いていないが，表を区切る縦線を入れたいならば，¥begin {tabular} {｜c｜r｜} のように書けばよい。2重線ならば‖だ。

表の中身は＆で区切り，1行の終わりは¥¥で宣言する。251行目の {¥it Rate No.} はRate No.をイタリックで書く指定。同行の¥hlineだが，これは単なる横線を入れる命令。もしも2重横線にしたいならば，¥hlineを2つ入れよう。

257～259行までは定義した環境のネストにすぎない。

6，7ページ目

たいしてすごいことはしていない。「Twentyone (c) Ext (T.Kawamoto)」が途中で切れたのは失敗だったかな(図11)。

いかにTeXが自動整形してくれるといっても，こういうのは自分で合わせるしかないという悪い例かな。

終わりに

以上。ドキュメントを書く程度ならば，多分この程度の知識で十分足りると思われる。

次回は，数式を中心に話を進めるので，理系大学生は実験レポートを溜めておいてほしい。

…………おい。

図8　ソースそのままを出力する

さて、以下は pcmplay.x の実際のヘルプです。

```
PCMPLAY.x $Id: pcmplay.c,v 1.21 1994/08/20 17:28:41 Kohju Exp $
                                         Presented by 瀬 康史

オンラインマニュアル
文法:
  pcmplay [-options:下記参照] filename [-options:下記参照] filename ....
説明:
  X680x0 の再生できる PCM を全てのレートで再生します.
  PPI ボードに接続された簡易8ビット PCM ボードでも再生可能です.
```

[2]一般に出回っているデータと言う訳ではないので、拡張子が定義されていないため、私が勝手に考えました

図9　右矢印を出力する

とコマンドラインから打ち込むことにより、再生することが出来ます.

拡張子がまったく指示されなかった場合、pcmplay.x が対応している拡張子を順にサーチしていきます.

これは以下の通りです.

$*.pcm \rightarrow *.p16 \rightarrow *.pc8 \rightarrow *.pn8 \rightarrow *.ps8 \rightarrow *.pce$

従って、ディレクトリに、*.pcm、*.p16、*.ps8 などが混在していた場合でも、

図10　節よりもさらに細かい区切り。表に注目

4.1 -R オプション

-R オプションは PCM の周波数レートを与えるオプションです.

ただし、周波数は任意に指定出来る訳ではなく、X680x0 が内示的に保有する周波数レートを指定します. これは、ヘルプに記載されている通り、0～4までで、以下の通りの周波数で再生されます.

表 1: -R で与える数字とサンプリングレートの対応

Rate No.	*Sampling Rate*
0	3.9kHz
1	5.2kHz
2	7.8kHz.
3	10.4kHz
4	15.6kHz

図11　単語が途中で切れていることに注意

そこで、pcmplay はセカンダリー拡張子を採用して、これらを省略することにしました. 省略した場合、-r オプションが指定されていない限り、15.6kHz で再生されます. セカンダリー拡張子は、Twenty-one(c)Ext(T.Kawamoto) が必要で、オプションとして+P(マルチピリオドモード) を指定する必要があります.

「X680x0 TeX」(ソフトバンク刊)の不具合について

「X680x0 TeX」に添付されているディスク内容に，不具合が報告されています。1.本文の記述とインストーラの出力メッセージが異なっている，2.全角スペースを含むソースをJXL4フォントで出力できない，3.X68030でBigTeXがバスエラーを起こすことがある，4.pLaTeXで作成したDVIファイルをデバイスドライバで出力できない場合がある，という4つです。1および2はすでに解決しましたが，3はまったくの原因不明です。4は本誌の先月号の記事によって不安定な要素があることがわかり，現在調査中です。3および4は，ソースあるいは環境によって発生する場合があるようで，必ず生じるというものではないようですから，一度は試してみることをお勧めします。なお，これらへの対応は，NIFTY/FSHARP3のほか，Sunday-Net(03-5467-5911)などのローカルネットでも行っています。　(ハードウェア活用書編集部)

リスト

```
  1: ¥documentstyle[a4j]{jarticle}
  2:
  3: ¥title{PCMPLAY Ver.1.21}
  4: ¥author{蕘焔手紅呪威羅黄爵神}
  5: ¥date{¥today}
  6:
  7: ¥begin{document}
  8: ¥maketitle
  9: ¥tableofcontents
 10: ¥listoftables
 11: ¥vfill¥eject
 12:
 13: ¥newpage
 14:
 15:
 16: ¥section{概要}
 17:
 18: このドキュメントは限定ドキュメントです。
 19:
 20: 連載のために、取捨選択しながら書き上げているため、大したこと
 21: は書いてありません。
 22:
 23: 本原稿は ¥LaTeX システムを利用することにより、読むことが出来
 24: ます。
 25:
 26: ¥section{使用許諾}
 27:
 28: pcmplay.xは無料です。今後、有料になる可能性はまったくありま
 29: せん。よって自由に使用し他人に譲ることもできます。ネットワー
 30: クや、BBS にアップロード、商的目的で使用することに関しては全
 31: く自由です。
 32:
 33: ただし、pcmplayは全くのフリーソフトウェアではありません。
 34: pcmplay は著作権法で保護されていますし、配布に関しては制限も
 35: あります。しかし、この制限は、善意の使用者の行動を禁じないよ
 36: うに定めることにします。禁じられているのは、第三者への譲与の
 37: 拒否、すなわち、pcmplayの普及を妨げることです。このような事
 38: をする人に、pcmplayを使う権利は有りません。
 39:
 40: pcmplayは無償で使用許諾されるので、著作権法ほかの日本国内の
 41: 法律で許される範囲での保証はいっさいしません。このドキュメン
 42: トに記載された資料においてもです。そしていかなる場合も、私、
 43: 蕘焔手紅呪威羅黄爵神及び、pcmplayを配布した第三者はpcmplayと
 44: そのアーカイブ¥footnote{この場合のアーカイブとはアーカイブの
 45: 形式まで制限している訳ではありません。また、アーカイブはオリ
 46: ジナルのまま、提供する必要はありませんが、明らかに害になるも
 47: のを含ませての配布は良心の上で行わないで下さい。} を、まった
 48: く保証なしに提供します。ここでいう保証とは、明示されたもの、
 49: 示唆されたものを問わず、市場性や特定の目的に対する適合性を持
 50: つ時に示唆される保証も含まれますが、それに限定されたものでは
 51: ありません。ソフトウェアの品質、性能などに関してのすべてのリ
 52: スクはすべて利用者が追うものとします。もし、pcmplay.xに欠陥
 53: があるとわかった場合、必要とされる修復処置、もしくは訂正に要
 54: する費用はすべて利用者が負担するものとします。この場合の費用
 55: の負担とは、 pcmplayにあたる正しい使い方をした時の欠陥による、
 56: 使用者の個人の管理するもの、使用者がかかわる公共に関するもの
 57: 損失による費用のことも意味します。 pcmplay本体のプログラムの
 58: 訂正などは、使用者が自由に行なってもかまいませんが、ある程度
 59: の修復にあたっては、私、蕘焔手紅呪威羅黄爵神がVersion upとい
 60: う形で修復を試みる場合があります。
 61:
 62: 法律で要求される場合を除き、私、蕘焔手紅呪威羅黄爵神と上記に
 63: よりpcmplayの修正と再配布を許されるすべての第三者はpcmplayを
 64: 使用、または使用できないことによる損失につき、いっさい責任を
 65: 持ちません。ここでいう損失とは、利益上の損失、金銭的な損失、
 66: pcmplayの使用あるいは、使用できないことに起因するそのほかの
 67: 偶発的なまたは間接的な損失が含まれます。データの損失、不正確
 68: に解釈された技術資料、第三者が負った損失、私、蕘焔手紅呪威羅
 69: 黄爵神以外の人物から配布された類似したソフトウエア、またはそ
 70: れらを改良したものなどとの非互換性なども含まれますが、これに
 71: 限定されることではありません。また、このような損失が起こる可
 72: 能性について利用者が知らされていた場合でも、あるいは別の人物、
 73: 団体からどのような申し立てがあった場合でも例外ではありません。
 74:
 75: ¥section{動作環境}
 76:
 77: この項では、ソフトの動作環境とインストールの方法を説明します。
 78:
 79: バイナリファイル、pcmplay.xは特に起動のための外部ファイルな
 80: どを必要としません。あなたがそのほかのツールをいれるディレク
 81: トリにpcmplay.xをcopyないしは、moveすることにより、インストー
 82: ルは終了します。pcmdrvやopmdrvの類は必要ありません。
 83:
 84: ¥section{使用目的と使用方法}
 85:
 86: この項では、ソフトの具体的な使用方法を説明します。
 87:
 88: このソフトは、基本的に各種のPCMデータを
 89: {¥gt 再生出来るようにする}目的で作られています。
 90:
 91: PCMのデータ形式は、非圧縮型、圧縮型を含め、多岐にわたりいろ
 92: いろありますが、現在のバージョンでは、次の種類しかありません。
 93:
 94: ¥begin{enumerate}
 95:     ¥item   MSM6258フォーマットのADPCM。拡張子はPCM。
 96:     ¥item   符号付き16bitPCM。拡張子はp16,phl。
 97:             (ビッグエンディアン)
 98:     ¥item   符号付き16bitPCM。拡張子はplh。
 99:             (リトルエンディアン)
100:     ¥item   符号付き 8bitPCM。拡張子はpc8,p8。
101:     ¥item   符号無し 8bitPCM。拡張子はpn8。
102:     ¥item   符号bit付き 8bitPCM。拡張子はps8。
103:             (MEGA CD、FM-TOWNS用)
104:     ¥item   謎の拡張子{¥em *}.PCEファイル。
105: ¥end{enumerate}
106:
107: 1は普通X680x0に内蔵されているMSM6858Vによるadpcmファイルです。
108: X680x0の世界では、*.pcmとされています。周波数レートは5種類
109: ありますが、これらは-rオプションで指定することができます。
110:
111: また、拡張子の前に、さらに拡張子を入れることにより、自動的に
112: 周波数レートを変えることのできるモードもあります。
113:
114: 2はビックエンディアンの16ビットPCMファイルです。
115: ZMUSICのボーナスディスクなどに多量に入っているので、よく
116: 使われるでしょう。視聴してみたいときなど便利でしょう。
117:
118: 3はPC9821やPC9801の86ボードの16bitPCMです。基本的には2の逆ワー
119: ドでしかありません。
120:
121: 4はPC9821やPC9801の86ボードの8bitPCMです。
122: $ -128(0x80) ¥sim +127(0x7f)$の範囲で、c言語の{¥em char}に
123: 相当しています。
124:
125: 5はPolyphone やOh!X誌94年6月号に掲載されたAD558Jボードの
126: データ形式符号無しのpcmです。
127: $0(0x00) ¥sim +255(0xff)$の範囲で、c言語の{¥em unsigned char}
128: に相当しています。中心点が$0x80,0x7f$なので注意が必要です。
129: ¥footnote{一般に出回っているデータと言う訳ではないので、
130: 拡張子が定義されていないため、私が勝手に考えました}
131:
132: 6番は主に、MEGA-CD、FM-Townsなどで使用されている8bitPCMです。
133:
134: $-128(0xff) ¥sim +127(0x7f)$の範囲です。MSBが符号ビットな為、
135: 中心点0が2つあります。$(0x00,0x80)$
136:
137: CDROMユーザーで、PCMファイルを集める習慣がある人なら、
138: これらのpcmデータが直接再生でき、
139: さらにはp16形式にまで変換できるこのツールは、
140: 非常にありがたいものになるでしょう。
141:
142: さて、以下はpcmplay.xの実際のヘルプです。
143:
144: {¥small
145: ¥begin{verbatim}
146:
147: PCMPLAY.x $Id: pcmplay.c,v 1.21 1994/08/20 17:28:41 Kohju E
xp $
148:                                            Presented by 瀧 康史

149:
150: オンラインマニュアル
151: 文法:
152:   pcmplay [-options:下記参照] filename [-options:下記参照] filen
ame ....
153: 説明:
154:   X680x0の再生できるPCMを全てのレートで再生します。
155:   PPIボードに接続された簡易8ビットPCMボードでも再生可能です。
156:
157: オプション:(大文字小文字は判別しない)
158:   Rn      サンプリングレートを指定します。nは0~6までの整数
159:           PCM8で使われているものとコンパンチブルで、下記の通りです。
160:           ADPCM(MSM6258V)
161:           0   3.9kHz   1   5.2kHz    2   7.8kHz    3  10.4kHz
  4  15.6kHz
162:   Pn      パンポットの指定。nは0~3までの整数で、下記の通りです。
163:           0   Non      1   Left      2   Right     3   Center
164:   Bn      内蔵PCMではなく、PPIボードに接続された簡易8ビットPCMで再生する。
165:           nはウエイト値の設定。デフォルトは128。
166:   O       ファイルに出力する。
167:   S       再生を中止します。
168:
169: PCM8が常駐しているので,PCM8モードで再生されます。
170: ¥end{verbatim}
```

▶付録ディスクがいまの私にとっていちばんの喜びです。こんな私っていったい……。
大仲　和彦(30)奈良県

```
171: }
172:
173: 基本的にpcmplayのオプション形式は法則があります。スペースを
174: 区切りに、ハイフンを利用して設定するのがオプションで、それ以
175: 外はファイル名です。従って、オプションの因子にスラッシュは使
176: えません。¥footnote{ルートからの絶対パス指定と混同するため}
177:
178: プログラムはHUPAIRに対応しているため、シェルがHUPAIR対応なら
179: ば、連続してデータを幾つでも渡すことが出来ます。また、シェル
180: にメタキャラクタを展開する機能がなくても、プログラム内部でOS
181: がサポートする範囲でのワイルドカード展開をするので、
182: command.xユーザーの人も違和感なく使うことができます。
183:
184: また、オプションは基本的に指定された後のファイルにしか影響し
185: ません。
186:
187: デフォルトのオプションモードは、拡張子によって決定されます。
188: 現在pcmplayで予約されている拡張子は、
189: *.pcm,*.p16,*.pc8,*.pn8,*.ps8で、これらはそれぞれ対応したデー
190: タフォーマットとして、内部に与えられます。
191:
192: 説明を省くために例を上げましょう。
193:
194: ¥begin{verbatim}
195: $pcmplay -p3 -r5 Center.pcm -p2 Right.p16 -p1 Left.ps8 -p3
-r2 Center.pcm
196: ¥end{verbatim}
197:
198: このようにオプションを与えた場合、Center.pcmはMSM6258Vの
199: ADPCM方式で周波数レートは5番の15.6kHz、パンポットセンターで
200: 再生されます。この再生が終わる¥footnote{再生を途中終了したい
201: ときにはSHIFT KEYを押します。}と、Right.p16を符号付き
202: 16BitPCM方式(15.6kHz)、パンポットライトで再生し、
203: その後、Left.ps8を符号ビット付き8BitPCM(15.6kHz)方式、
204: パンポットレフトで再生します。
205: 最後にパンポットセンターで、最初のCenter.pcmを周波数レート
206: 2番の7.8kHzで再生する訳です。
207:
208: 上記のように意味の重なるオプションは、次に相殺されるオプショ
209: ンが出現されるまで持続されます。当然ながら、ディレクトリには
210: *.p16,*.pcmなどが混在していた場合でも、
211:
212: ¥begin{verbatim}
213: $pcmplay *.p16 *.pcm
214: ¥end{verbatim}
215:
216: とコマンドラインから打ち込むことにより、再生することが出来ま
217: す。
218:
219: 拡張子がまったく指示されなかった場合、pcmplay.xが対応してい
220: る拡張子を順にサーチしていきます。
221:
222: これは以下の通りです。
223:
224: ¥[{¥em  *.pcm ¥rightarrow *.p16 ¥rightarrow *.pc8
225:  ¥rightarrow *.pn8 ¥rightarrow *.ps8 ¥rightarrow *.pce}¥]
226:
227: 従って、ディレクトリに、*.pcm、*.p16、*.ps8などが混在してい
228: た場合でも、
229:
230: ¥begin{verbatim}
231: $pcmplay *
232: ¥end{verbatim}
233:
234: と再生するだけで、すべてのPCMファイルを順に、適当なPCMモード
235: に可変しながら再生します。
236:
237: ¥subsection{-Rオプション}
238:
239: -RオプションはPCMの周波数レートを与えるオプションです。
240:
241: ただし、周波数は任意に指定出来る訳ではなく、X680x0が内示的に
242: 保有する周波数レートを指定します。これは、ヘルプに記載されて
243: いる通り、$0 ¥sim 4$までで、以下の通りの周波数で再生されます。
244:
245: ¥bigskip
246:
247: ¥begin{table}[hbtp]
248: ¥caption{-Rで与える数字とサンプリングレートの対応}
249: ¥begin{center}
250: ¥begin{tabular}{cr} ¥hline
251:  {¥it Rate No.} & {¥it Sampling Rate} ¥¥ ¥hline
252:  0 & 3.9kHz ¥¥
253:  1 & 5.2kHz ¥¥
254:  2 & 7.8kHz ¥¥
255:  3 & 10.4kHz ¥¥
256:  4 & 15.6kHz ¥¥ ¥hline
257: ¥end{tabular}
258: ¥end{center}
259: ¥end{table}
260:
261:
262: ¥bigskip
263:
264: 拡張子がpcmであるファイルは、X680x0ではadpcmファイルですが、
265: このデータ形式にはヘッダが無いため、サンプリングレートをファ
266: イルに包含することは出来ません。従って、15.6kHz以外のレート
267: であるデータは、後に記すセカンダリー拡張子を使用しない限り、
268: -Rオプションによって記述せねばなりません。
269:
270: ¥subsection{-Pオプション}
271:
272: パンポットの指定を行います。
273:
274: 引き数は、そのままIOCSコール、またはpcm8コールに渡されますが、
275: ただpcmを再生するのに右、左などを指定する必要などはないでしょ
276: うから、まずつかわれないオプションでしょう。
277:
278: ¥subsection{-Bオプション}
279:
280: 拡張ボードで再生する指定です。
281:
282: とりあえず付けているので、大したことはできません。
283:
284: ¥subsection{-Sオプション}
285:
286: 割と便利なのがこのオプションなのですが、実は、pcm8が常駐して
287: いないと使えません。
288:
289: 効果はpcmの再生を停止するだけです。
290:
291: pcmplay -sを、pstpなどとエイリアスしておき、なにかぐちゃぐちゃ
292: で長いものを再生してしまったとき、pcmを止めるのに使うと良い
293: でしょう。
294:
295: ¥subsection{-Oオプション}
296:
297: ps8,pn8,pc8などの形式のファイルで、周波数が15.6kHzでなかった
298: 場合、このモードでは2つのファイルを吐き出します。ひとつはそ
299: のままの周波数でのp16ファイルを、もうひとつは、15.6kHzに変換
300: したp16ファイルです。もちろん、これらのファイルがもともと
301: 15.6kHzだった場合、出力するファイルは、15.6kHzのp16ファイル
302: のみです。
303:
304: p16でも周波数が15.6kH以外だった場合、15.6kHzに変換してファイ
305: ルに出力します。
306:
307: このようなとき、出力ファイルはセカンダリー拡張子に影響される
308: ので、twentyoneが常駐していないとなりません。
309:
310: ¥subsection{セカンダリー拡張子}
311:
312: PCMは基本的に、PCMフォーマット、そして周波数レートさえわかれ
313: ば、完璧に再生できます。しかし、フォーマットは拡張子によって
314: 得ることができますが、周波数まではできません。
315:
316: そこで、pcmplayはセカンダリー拡張子を採用して、これらを省略
317: することにしました。省略した場合、-rオプションが指定されてい
318: ない限り、15.6kHzで再生されます。セカンダリー拡張子は、
319: Twentyone(c)Ext(T.Kawamoto)が必要で、オプションとして+P(マル
320: チピリオドモード)を指定する必要があります。
321:
322: 設定方法は具体的には、拡張子の前に、拡張子を付けることにより
323: 記述します。すなわち、周波数レート3、10.4kHzのpcmを再生した
324: い場合、ファイル名がfoo.pcmならば、foo.3.pcmとあらかじめリ
325: ネームしておくことにより、pcmplay.xが内部でオプション展開を
326: し、10.4kHzで再生を行います。
327:
328: *.pcm形式以外の場合は、$10^{-2}$倍した周波数を直接書き込むこ
329: とによって実現できます。たとえば、44.1kHzのp16のPCMを再生し
330: たい場合、foo.441.p16とすることにより、内部で自動的に周波数
331: 変換をして再生できます。
332:
333: ただし、これには時間がかかるのでご了承ください。
334:
335: ¥section{今後の予定と、現在発覚している不都合}
336:
337: 実は現在発見されている不都合は、いろいろありまして、この理由
338: は単に私の忙しさと、それに起因する怠慢から来ています。
339:
340: ソースにリヴィジョンの歴があり、これを読んだ方はわかると思い
341: ますが、bgdrvに対応していると書いてあるのに、まともに対応し
342: ていません。
343:
344: 結局、行ったのはpcm8の占有程度でして、pcmの途中演奏解除機能
345: をBITSNSで行っているという事実から、bgdrvのバックグラウンド
346: タスクでpcmplayを走らせた場合、表タスクでSHIFTキーを押すだけ
347: で、pcmファイルの演奏を停止してしまいます。
348:
349: これを解決する方法はありますが、単に「面倒である」という要因
350: からやっていません。コマンドラインから、$pcmplay *.pcm *.p16
351: > nul $¥&とやると、バックグラウンド再生ができて面白いのです
352: が。
353:
354: さらに、セカンダリー拡張子関連で、twentyoneが無い場合に、ア
355: ウトプットモードを利用すると、どうなるかわかりません。(多分
356: 飛ぶ)
357:
358: 今後は、これらのことを直さなくてはいけません。
359:
360: 新しく機能を追加するならば、ヘッダのついたサンプリング
361: ファイルのヘッダ分析ルーチンなどは付けてみようかと思っていま
362: す。(Windows方面や国民機, FM-Towns方面のpcmフォーマットなども)
363:
364: ¥section{連絡先}
365:
366: バグ、要望があった場合、次の方法のいずれかの方法で連絡して
367: ください。連絡がつきやすい順に、
368:
369: ¥begin{enumerate}
370:     ¥item   PCVAN-ID X1CLUBに電子メールを出す。
371:     ¥item   PCVAN内SIG X1CLUBのソフトウェアに書き込む。
372:     ¥item   Oh!X編集部内、瀧宛に郵政省メールで、手紙を出
373: す。
374: ¥end{enumerate}
375:
376: おそらくいずれかの方法で捕まるはずです。
377:
378: ¥end{document}
```

▶某アニメデータを作るためにX68000初代とイメージユニットを買いました。ディスプレイの横にはツインタワーが2つそびえ立っています。　小川　克仁(20)愛知県

Oh!X

LIVE in '94

X68000・Z-MUSIC ver.2.0用(SC-55対応)

©Falcom「イース2」より

Tender People

Sasaki Shinya 佐々木 信也

X68000・Z-MUSIC ver.2.0用

©KONAMI MSX用「GRADIUS2」より

エンディングテーマ

Shindo Noriyuki 進藤 慶到

X68000・Z-MUSIC用(CM-64対応)

NATURE

Fukui Yuki 福井 祐貴

全国的な熱暴走の危険を誘うような猛暑も過ぎ去り，X68000にもつらい季節が終わりを告げたようですね。人間もパソコンも頭冴え冴えでしょうか。ひと息ついたところで，今月も力のこもった3曲です。CM-64も久しぶりの出番ですよ。

以前予告したZ-MUSICツール本は思うように進行していないらしいです。皆さん，はがきで担当の人をつつきましょう(あたたかい励ましの声をお待ちしています：担当者)。というわけで，今月は私・西川善司がこのコーナーの進行役を務めます。さて1曲目は……。

ラミア村へ行こう

ファルコムの黄金時代を象徴する「イース2」。今月の1曲目は，この「イース2」のラミア村のテーマ「Tender People」のアレンジバージョンです。

演奏には，SC-55が必要です。

佐々木君がこの曲を作ろうとしたのは，スーパーファミコンのゲーム「ロマンシング サ・ガ」のCDを聴いたのがきっかけとのこと。原曲のイメージを壊すことなく膨らませた見事なアレンジセンスに深い感銘を受け，今回のこの「Tender People」のアレンジも，そのCDのアレンジ技法に相当な影響を受けているそうです。なるほど，まずイントロからも単なる原曲のグレードアップバージョンでないことは伝わってきます。リズムパートのティンパニや原曲にはないオブリガードフレーズも見え隠れし，なかなかプロっぽいサウンドに仕上がっています。

再現！ MSX版「GRADIUS2」

2曲目はお馴染みの進藤君の作品で，MSX版「GRADIUS2」よりエンディングテーマです。

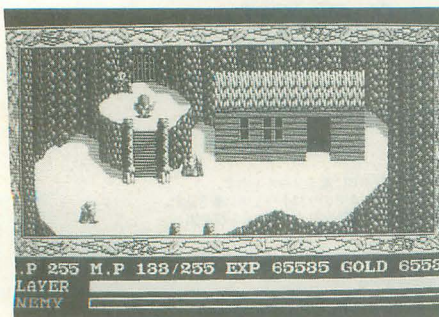
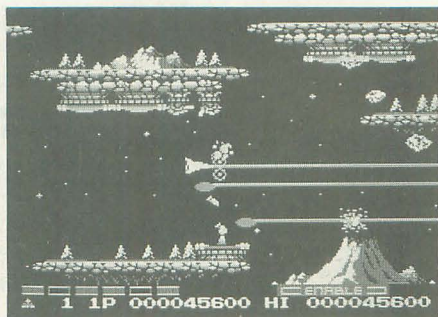
昨年の11月にSPSから発売されたX68000用ゲーム「ネメシス'90改」は，このMSX版「GRADIUS2」をX68000用にグレードアップ移植したものですが，そのBGMはアレンジか，はたまた音の間違い(バグ)かと，その評価についてはファンの間でかなり激しい議論が交わされました。進藤君はその「ネメシス'90改」の音楽には相当腹を立てた人間のひとり。そこで彼は，温故知新も兼ねてMSX版「GRADIUS2」の曲のコピーを始めたというわけです。オリジナルはSCC音源というPSGに簡単な音色作成機能のついた独特の音色を発する音源だったのですが，このSCC音色のニュアンスをZ-MUSICのモジュレーション機能などを駆使して見事に再現しています。

X68000本体内蔵FM音源のみで構成されているので，AD PCMデータやMIDIモジュールは一切不要です。リストもすっきりした構成で，初心者にも入力しやすいでしょう。原曲を知る人も知らない人も，ぜひ入力して聴いてみてくださいね。

イメージは遙かに

最後は，福井裕貴君のオリジナル曲「NATURE」をお届けしましょう。曲は無限ループ構成で，対応楽器はCM-64です。

空を飛んでいるイメージで作曲したところ，友人から水の流れを想像するといわれたので，タイトルに「自然」とつけたそうです。さて，読者の皆さんはこれを聴いて何を連想するでしょうか。

福井君は「魔女の宅急便」や「となりのトトロ」で著名な作曲家・久石譲のファンだそうで，「曲の構成などは久石氏のアルバム『インフォメーション』の影響を受けている」とのこと。なるほど，イントロから始まるアルペジオパターンやコードが奏でる独特のハーモニーは久石氏によるNHK「人体」をイメージさせてくれます。

イース2

GRADIUS2

リスト1 イース2

```
  1: .comment - Ys2 - Tender People - SC55(GS) - Arr by N.Imaneko
  2: /                    !!  Special Thanks   Darios·Sawm !!
  3:
  4: / INIT
  5:
  6: (i)
  7: (b1)
  8: .SC55_INIT $10
  9:
 10: /        TRACK SET
 11:
 12: (m1,4100)(aMIDI1,1)       /
 13: (m2,2100)(aMIDI2,2)       /      - YS2 - Tender People -
 14: (m3,2100)(aMIDI2,3)       /
 15: (m4,2100)(aMIDI4,4)       /For ZmusicSystem Ver2 & Roland SC55
 16: (m5,2100)(aMIDI5,5)       /
 17: (m6,2100)(aMIDI6,6)       / - Arr by N.Imaneko 1994/05/27/Fri-
 18: (m8,2100)(aMIDI8,8)       /
 19: (m9,2100)(aMIDI9,9)       /        - Version 1.0 - X -
 20: (m10,2100)(aMIDI10,10)    /
 21: (m11,2100)(aMIDI11,11)    /
 22: (m12,2100)(aMIDI12,12)    /
 23: (m13,2100)(aMIDI13,13)    /
 24:
 25: / DRAM PART SET
 26:
 27: .ROLAND_EXCLUSIVE $10,$42={$40,$1A,$15,$02}
 28: .ROLAND_EXCLUSIVE $10,$42={$40,$1B,$15,$02}
 29: .ROLAND_EXCLUSIVE $10,$42={$40,$1C,$15,$02}
 30:
 31: / SOUND SET
 32:
 33: .ROLAND_EXCLUSIVE $10,$42={$40,$10,$35,$55}
 34:             /Part10:Decay:( $0E - $72 )
 35:
 36: .SC55_V_RESERVE = {3,4,0,2,2,1,0,2,2,2,3,1,2,0,0,0}
 37:
 38: / EFFECT
 39:
 40: (t1)       @e75,20/guitar
 41: (t2)       @e127,20/harmonica.
 42: (t4)       @e80,40/brass
 43: (t5)       @e80,40
 44: (t6)       @e126,20
 45: (t8)       @e100,0/bass
 46: (t9)       @e100,0
 47: (t10)      @e127,20/parc
 48: (t11)      @e50,20
 49: (t12)      @e10,20
 50: (t13)      @e110,20
 51:
 52: / MML (18 18 18)/4.
 53:
 54: / +--- guitar --------+
 55:
 56: (t1)       @26 v12 @u100 o4 q8 14 t160 r1 /
 57:
 58: / intoroduction 1,2,3,4
 59:
 60: |:|:@u114'c8>a'>@u100'bf' <<c>b8 'c8>a'>@u106'be'
 61: <{@u115'ec'@u110'd>a'@u100'c>a'}4.
 62: @u114'c8>g'>@u100'bg' <<c>b8 'c8>a'>@u106'be'
 63: <{@u110'ec'@u100'd>a''c>e'}4.
 64:
 65: @u110'c8>a'>@u100'bf'
 66: @u90<<c>b8 @u118'c>g'>@u110'b8g'@u120'ae'@u100'b8g'
 67: <{@u120'c>g'>@u100'c>a''e>b'}4. {@u120'gc'@u100'be''<c>g'}4.
 68: {@u120'e>b'@u100'gc''be'}4. {@u120<'d>b'@u100>'bg''ge'}4.
 69: @u100 :|<v11:|
 70:
 71: / [ a ]
 72:
 73: [$]18 v10
 74: z110,115,120,108,100,110,110,115
 75: 'fd''f4d'>'af'<'d>b''fd''a4.f'r'd>b''fd'
 76: z110,112,100,105,110
 77: 'ge''g4e''d>b''d>b''d>b'>'b4g''bg''ge''ge''ge'
 78: z120,100,100,120,110,100,110,108,106,105
 79: <'c>a''c>a'>'e>c''ge''b4g'<'c>a'>'g4e''bg'<'c>a''c>a'
 80: z120,100,98
 81: 'ec''ec''c>a'>'ge''ge'<'c>a'>'ge''ge''ec''af''af'<'c>a'
 82:
 83: z @u110 'fd''f4d'>'af'<'d>b''fd''a4.f'r'd>b''fd'
 84: 'g4e''d>b'r<'ge'>'ge'r<'ge'>'ge''ge''g4e'
 85: <'c4>a''c>a'>z127,110,103
 86: 'c>a''ec''ge''c>a'<'c>a'>'c>a'<'c>a'>'ge''ec'
 87: 'ec'<'ec'>'c>a''ge'<'ge'>'ec''c>a'<'c>a'>'d>b''ec'<'ec'>'c>a'
 88:
 89: / [ b ]
 90:
 91: 14. @u110 |:4'af':||:4'bg':||:4'ge':||:4'bg':|
 92: |:4'af':||:4'cg#':| |:4'ec':||:4'ag':|
 93: |:4'af':||:4'g#f':|18 |:4'c4e''ce'<'ce'>'ce''ce':|[*]
 94:
 95: v12 14 |:
 96: @u110'c8>a'>@u100'bf' <<c>b8 'c8>a'>@u106'be'
 97: <{@u115'ec'@u110'd>a'@u100'c>a'}4.
 98: @u110'c8>g'>@u100'bg' <<c>b8 'c8>a'>@u106'be'
 99: <{@u110'ec'@u100'd>a''c>e'}4.
100:
101: @u110'c8>a'>@u100'bf'
102: @u90<<c>b8 @u118'c>g'>@u110'b8g'@u120'ae'@u100'b8g'
103: <{@u120'c>g'>@u100'c>a''e>b'}4. {@u120'gc'@u100'be''<c>g'}4.
104: {@u120'e>b'@u100'gc''be'}4. {@u120<'d>b'@u100>'bg''ge'}4.
105: @u100 :|
106:
107: [d.s.]
108: [coda]
109:
110: @u120‾2r4.r4'c8eb',6'a2.eb',0 & ¥32'aeb'*576
111:
112:
113: / +--- harmonica -----+
114:
115: (t2)       @23 v12 @u100 o4 q8 14. @j1 @p52
116: (t3)                @u120 o3 q8 14. @j1
117: (t2,3) r1
118:
119: |:7 rrrr :| rrr   {g&a&b&<c&d&e&f}
120:
121: |:z100,80,85,90,100,90 g2.&{gab}2. g2.e4.
122: >z100,95,90,85,80,90 {g&a&b&<c&d&e&f}
123: @u100 g2.&{@u95ga@u97b}2. @u105<c4>a8g8&g4&g2e4
124:
125: z100,110,100,110,112,115 g4e8g2.{g&a&b} @u110 g4e8g2.{g&e&c}
126: z110,115,105 >a2.<e4.d4. z97,87,91
127: {dcdcdcz90,80,85dcdcz80,70,75dcdcz85,75,80dcdcdcz70,60,75dcdcd
cdcdcdc}1.
128:
129: / [ a ]
130:
131: |:8 rrrr :|
132:
133: / [ b ]
134:
135: 18 _2 z100,110,100,110,100,110,100 f2.>ab<cdef
136: z100,110 g4>fg4b<d4c>b4<c
137: z95,110,100,113,112,102,110,110,100
138: e4>b<g4eb4g&gef g4>b<d4c>b4<d&dc>b
139: @u110<a2.@u100c4.d4e g#2.d4eg4<c >b4ga4eg4b&b<d>b <e4.c2.cde
140:
141: @u105|:c4.&c2.ede:|@u110 c1.&c1.
142:
143: q4_1|:4r1.:|>@22 z100,80,80,110,105,100,100,75,75,75
144: |:4 ggg rgg ggr ggg :| ‾3q8
145:
146: / [ a' ]
147:
148: 14. >v14 z80,90 ff z110,105,100 de4f8 @u100g z102,104,106 ece
149: z105,110,100,110,115 ec<c>g4f8 q8
150: |:z110,112,110,117,110g4g8g8b8a8:|
151: @u125 a2.@u110 fg4a8 ggbg 18 z95,91,90,88,88 <c4cccc>b4bbbb
152: a4aaaag4gggg<
153:
154: / [ b' ]
155:
156: 18  _3 z115,110,100,110,100,110,100 f2.>ab<cdef
157: z100,110 g4>fg4b<d4c>b4<c
158: z95,110,100,113,112,102,110,110,100 e4>b<g4eb4g&gef
159: g4>b<d4c>b4<d&dc>b
160: @u110<a2.@u100c4.d4e g#2.d4eg4<c >b4ga4eg4b&b<d>b <e4.c2.cde
161:
162: @u106|:c4.&c2.ede:|@u110 c1.&c1. r4.@u90>r4{ceg} b2.& ¥32b*576
163:
164: / +--- brass ---------+
165:
166: (t4)       @61 v14 @u100 o4 q7 14. @j1       @p79
167: (t5)       @61 v10 @u100 o4 q7 14. @j1 @k4 r8 @p79
168: (t6)       @58 v11 @u100 o3 q7 14. @j1       @p79
169: (t4,5,6) r1
170:
171: |:16 rrrr:|
172:
173: / [ a ]
174:
175: z124,115,110,110,115,125a2.&a2&a8{f&g&a}8 z120,115,110,105bged
176: z118,105,110e2.&ed z120,100,90,100,105,120c&c4{c&d&d#}8q8e2.q7
177:
178: z110,110,110,110,115,120a2.&a2&a8{f&g&a}8z120,115,110,105bg<ed
179: >g2.&g&g4 z100,99,98,97,96,95,94,90 {gab<cdef}8 @u115 q8g1.
180:
181: / [ b ]
182:
183: ‾1z115,100,100,98,95 >f2.&f{def} @u100 g2.ba e1. @u95 g1.
184: z100,110,100,90 a2.cd4e8 @u100 g#2.
185: d4e8g4<c8 z100,95,98,100,105 >g2.<ge c>a2.&a z
186:
187: a2.&aa8b8<c8 >g#2.&g#g#8b8<c8 >e1.&c1.
188:
189: (t6)       <v13 @62
190: (t4,5,6) _7 q4 z100,100,100,80,80,80,70
191: |:7g8r4 g8r4 r8g8e8 e8e8e8:| g8r4 g8r4 r8g8e8 ‾6q8
192: @u124 r4r16{e&f&g}16
193:
194: / [ a' ]
195:
196: z124,115,110,110,115,125 a2.&a2&a8{f&g&a}8
197: z120,115,110,105 q7 bged q8
198: z118,105,110 e2.&ed
199: z120,100,90,100,105,120c&c4{c&d&d#}8q8e2.q7
200:
201: z110,110,110,110,115,120a2.&a2&a8{f&g&a}8z120,115,110,105bg<ed
202: >g2.&g&g4 z100,99,98,97,96,95,94,90 {gab<cdef}8 @u115 q8g1.
203:
204: / [ b' ]
205:
206: ‾1z115,100,100,98,95 >f2.&f{def} @u100 g2.ba e1. @u95 g1.
207: z100,110,100,90 a2.cd4e8 @u100 g#2.
208: d4e8g4<c8 z100,95,98,100,105 >g2.<ge c>a2.&a z
209:
210: a2.&aa8b8<c8 >g#2.&g#g#8b8<c8 >e1.&e1
211:
212: @u127‾1 r2.>g2.& ¥32g*576
213:
214:
215: / +--- bass ----------+
216:
217: (t8)       @34 v14 @u110 o3 q7 14.
218: (t9)       @33 v13 @u95  o3 q7 14.
219:
220: / intoroduction 1,2,3,4
221:
```

```
222: (t8,9)    r1
223: rrrr rrrr rrrr rrr 18 >r(b4>b)<
224:
225: |:3d4.{>a<df}4. a4.{>gab}4. <c4.{ec>b}4. a4.{ab<c}4.
226: d4.{>a<df}4. a4.{fd>b}4. <c4.{>b<ce}4. c4.{ggg}4.:|
227:
228: / [ a ]
229: [$]
230: |:|:d4.{>a<df}4.:| |:g4.{>gab}4.:| | |:<c4.{>efg}4.:|
231: |:a4.{ab<c}4.:|:| 14. |:{ccc}:|>|:{bbb}:| |:{aaa}:||:{ggg}:|
232:
233: / [ b ]
234:
235: f4f8{fff} f4f8{<fc>a} g4g8{ggg} g4g8{<gd>g}<
236: c4c8{ccc} c4c8{>b<cd} e4>a8{aaa} a4a8{a<ag}
237:
238: 18 f4f4ff f4f4ff f4f4ff fff4ff
239: c4cccc c4ceee >a4aaaa a4a<a>g<g
240:
241: >|:4f4f<f>ff:| <|:4c4c<c>cc:|>[*]
242:
243: |:<d4.{>a<df}4. a4.{>gab}4. <c4.{ec>b}4.  a4.{ab<c}4.
244:    d4.{>a<df}4. a4.{fd>b}4. <c4.{>b<ce}4. c4.{ggg}4.:| [d.s.]
245:
246: [coda]
247:
248: ‾1r4.(e4>e)rc2.& ¥32c*576
249:
250: / +--- parc ----------+
251:
252: (t10)     @49 v10 o3 q7 14. @p64 r1 k1 /
253:
254: /into
255: rrrr rrrr rrrr rrrr
256: k0 18<@u96 |:4 d-4c d-4. rcc | d-d-d-:| @u90<b4.@u70[$] @u93
257: |:4a4a-a4a-ra-a-a4a- a4a-a4a-|aa-a-aa-a-:| |:aa-16a16a-:|
258:
259: @e40,105 @p102
260: v12 @u90 14.@k12 |:r>>>a<d>a @k10b4.@k12@u100 18k1cccc4cccck0
261: @u90 g4.@u100
262: 18c#c#c#@u90 <@k10c4.>@u100c#c#c# @u9014.@k11rggg:| @k0
263: @e127,20 @p64
264: <<<18 v8 |:12 @u97ref efe ref @u92116ee er ee18@u97 :|v11
265:
266: [d.s.]
267:
268: r4. b4. b4. r1.
269:
270: (t11)     @33 v13 o2 q8 @u100 14. @p80 r1 /bd sd cymbal
271:
272:   /into
273: rrrr rrrr rrrr rrr18ccc
274: |:3'c<c#a'ce cce cce cce  cce c16c16ce eec ce16e16e
275: cce cc16c16e cce eee  cce cc16c16e cec e16e16e16e16e16e16:|
276:
277:   /[a][b]
278: [$]'c<c#a'ce cce cce cee16e16  cce cce cce c'c<g'e
279: 'c<c#a'ce cce cce cee ece cec cec 'c<g''c<g''ce<g'@u100
280: |:3'c<c#a'ce cce cce cce  cce c16c16ce eec ce16e16e
281: cce cc16c16e cce eee cce cc16c16e cec e16e16e16e16
282: 'e16<g''e16<g':|
283:
284:   /[tuna1]
285: |:'c<a'ceccecce|cce:| 'e<g''e<g''e<g'
286: cce cc16c16e cce eee  cce cc16c16e cec e16e16e16e16e16e16[*]
287:
288:   /[tuna2]
289: |: 'c<c#a'ce cce cce cce  cce c16c16ce eec ce16e16e
290: cce cc16c16e cce eee  cce cc16c16e cec e16e16e16e16e16e16:|
291:
292: [d.s.]
293: [coda]
294:
295: @u127v14'c4e'c 'c4.e' 'c<c#a'*576
296:
297: (t12)     @33 v10 o2 q8 @u90 14. @p64 k1 r1 /hihat tom
298:
299:   /into
300: rrrr rrrr rrrr rrrr 18
301: |:11 z100,80,80 fff z100,70,90 faf @u80a@u90f16f16f16f16 aff:|
302: z100,80,80 fff z100,70,90 faf
303: k0 @u115 o2 b16b16b16b16a16a16 a16a16g16f16f16f16 k1 o2
304:
305:   /[a][b]
306: [$]|:7z100,80,80fff z100,70,90faf @u80a@u90f16f16f16f16 aff:|
307: z100,80,80 fff z100,70,90 faf
308: k0 @u115 o2 b16b16b16b16a16a16 a16a16g16f16f16f16 k1 o2
309: |:7 z100,80,80 fff z100,70,90 faf @u80a@u90f16f16f16f16 aff :|
310: z100,80,80 fff z90,80,80 aff @u90 f16f16f16f16f16f16 @u120 aaa
311:
312:   /[tuna1]
313: |:4 z127,100,95 fff z100,80,90 | afa :| ffa
314: z100,80,80fff z100,70,90 faf @u80a@u90f16f16f16f16 aff
315: z100,80,80fff z90,80,80 aff @u90 f16f16f16f16f16f16 @u97aaa[*]
316:
317:   /[tuna2]
318: |:|:3z100,80,80 fff z100,70,90 faf @u80a@u90f16f16f16f16 aff:|
319: z100,80,80 fff z100,70,90 faf
320: k0 @u115 o2 b16b16b16b16a16a16 a16a16g16f16f16f16:| k1o2
321:
322: [d.s.]
323: [coda]
324:
325: @u127r4.o2f-4.r2. r*576
326:
327: (t13)     @33 v12 o3 q8 @u107 14. @p64 r1 /other
328:
329: |:15 rrrr :| rrrb [$]|:8 bbbb :|@u103|:32 d#4d#8 :|
330: |:16d#4b8:|[*]|:32 b4b8:|[d.s.]
331:
332: [coda]
333:
334: r4.r418b ‾2 z120,110,100,90,120,100 bbbbbb @u127b*576
335:
336: (p)
```

リスト2 イース2のカウンタ表示

```
 1:00004C20 00000000   2:00004C20 00000000   3:00004C20 00000000   4:00004C20 00000000
 5:00004C38 00000000   6:00004C20 00000000   8:00004C20 00000000   9:00004C20 00000000
10:00004AB8 00000000  11:00004B90 00000000  12:00004C20 00000000  13:00004C20 00000000
```

リスト3 GRADIUS2


```
 1: .comment -GRADIUS2- END TITLE (C)KONAMI  by ENG 94/01
 2:
 3: / for ZMUSIC.X
 4:
 5: /-----------------------------------------------------------
 6: / TRACK SETUP
 7:
 8: (i)
 9:
10: (m1,3000)(aFm1,1)
11: (m2,3000)(aFm2,2)
12: (m3,3000)(aFm3,3)
13: (m4,3000)(aFm4,4)
14: (m5,3000)(aFm5,5)
15: (m6,3000)(aFm6,6)
16: (m7,3000)(aFm7,7)
17: (m8,3000)(aFm8,8)
18:
19: /-----------------------------------------------------------
20: / OPM TONE DATA
21:
22: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
23: (@1,      31,  0,  0, 15,  0, 24,  0,  2,  0,  0,  0
24:           31, 21, 13, 15,  3, 32,  0,  5,  0,  0,  0
25:           31, 21, 13, 15,  3, 25,  0,  2,  0,  0,  0
26:           29, 17, 13, 15,  2,  0,  0,  1,  0,  0,  0
27: /         AL  FB
28:            5,  7)
29:
30: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
31: (@2,      31,  0,  0,  0,  0, 24,  0,  2,  0,  0,  0
32:           31, 21,  1,  9,  3, 35,  0,  5,  0,  0,  0
33:           31, 21,  1,  9,  3, 28,  0,  2,  0,  0,  0
34:           31, 17,  1,  9,  2,  3,  0,  1,  0,  0,  0
35: /         AL  FB
36:            5,  7)
37:
38: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
39: (@3,      31,  0,  0,  0,  0, 25,  0,  1,  0,  0,  0
40:           31,  0,  0,  0,  0, 15,  0,  6,  7,  0,  0
41:           31,  0,  0,  0,  0, 65,  0,  1,  1,  0,  0
42:           21, 15,  0, 10,  1,  2,  0,  1,  0,  0,  0
43: /         AL  FB
44:            2,  7)
45:
46: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
47: (@5,      28,  0,  0,  0,  0, 25,  0,  1,  0,  0,  0
48:           28,  0,  0,  0,  0, 15,  0,  6,  7,  0,  0
49:           28,  0,  0,  0,  0, 65,  0,  1,  1,  0,  0
50:           26, 13,  0,  9,  2,  1,  0,  1,  0,  0,  0
51: /         AL  FB
52:            2,  7)
53:
54: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
55: (@6,      31,  0,  0,  0,  0, 25,  0,  1,  7,  0,  0
56:           31,  0,  0,  0,  0, 15,  0,  6,  7,  0,  0
57:           31,  0,  0,  0,  0, 65,  0,  1,  7,  0,  0
58:           28, 20,  0,  8,  2,  0,  0,  1,  7,  0,  0
59: /         AL  FB
60:            2,  7)
61:
62: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
63: (@7,      31,  0,  0,  9,  0, 15,  0,  9,  3,  0,  0
64:           30, 18,  0,  9,  2, 16,  0,  4,  7,  0,  0
65:           30, 18,  0,  9,  2, 12,  0,  4,  1,  0,  0
66:           30, 18,  0,  9,  2,  9,  0,  1,  5,  0,  0
67: /         AL  FB
68:            6,  0)
69:
70: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
71: (@8,      31,  0,  0,  0,  0, 26,  0,  1,  7,  0,  0
72:           26, 15, 10, 14,  3,  3,  0,  1,  7,  0,  0
```

▶うちの子へのメッセージ。たのむから勝手にディスクを吐き出さんでくれ〜。
原田　健史(27)東京都

```
 73:            31, 11,  0,  8,  4, 40,  0,  5,  3,  0,  0
 74:            31,  0, 15, 14,  0, 13,  0,  8,  3,  0,  0
 75: /          AL  FB
 76:             4,  7)
 77:
 78: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 79: (@9,       31,  0,  0,  0,  0, 14,  0, 10,  3,  0,  0
 80:            26,  0,  0,  7,  0,  8,  0,  4,  7,  0,  0
 81:            26,  0,  0,  8,  0,  6,  0,  4,  3,  0,  0
 82:            26,  0,  0,  8,  0,  4,  0,  1,  0,  0,  0
 83: /          AL  FB
 84:             6,  0)
 85:
 86: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 87: (@10,      31,  0,  0,  0,  0, 16,  0,  1,  7,  0,  0
 88:            23,  8,  1,  9,  1, 17,  0,  1,  7,  0,  0
 89:            31,  0,  0,  0,  0,  0,  0,  1,  3,  0,  0
 90:            23,  8,  1,  9,  1, 22,  0, 12,  3,  0,  0
 91: /          AL  FB
 92:             4,  5)
 93:
 94: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 95: (@11,      31,  0,  0,  0,  0, 25,  0,  1,  0,  0,  0
 96:            28, 20,  0,  8,  2,  6,  0,  1,  0,  0,  0
 97:            14, 10, 11,  8,  4, 27,  0,  4,  3,  0,  0
 98:            12, 10, 11,  8,  4, 26,  0,  1,  7,  0,  0
 99: /          AL  FB
100:             6,  7)
101:
102: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
103: (@12,      31,  0,  0,  0,  0, 27,  0,  1,  1,  0,  0
104:            31,  0,  0,  0,  0, 15,  0,  5,  1,  0,  0
105:            31,  0,  0,  0,  0, 65,  0,  1,  1,  0,  0
106:            21, 17,  0, 10,  1,  4,  0,  1,  1,  0,  0
107: /          AL  FB
108:             2,  7)
109:
110: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
111: (@13,      31,  0,  0,  0,  0, 15,  0,  1,  7,  0,  0
112:            18, 12,  2,  9,  1, 18,  0,  1,  0,  0,  0
113:            31, 14,  0,  9,  1, 30,  0,  4,  7,  0,  0
114:            31, 14,  0,  9,  1, 32,  0,  1,  3,  0,  0
115: /          AL  FB
116:             6,  5)
117:
118: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
119: (@14,      31,  0,  0,  0,  0, 15,  0,  1,  7,  0,  0
120:            24, 12, 12,  7,  1, 18,  1,  1,  0,  0,  0
121:            24, 17, 11,  8,  2, 24,  0,  4,  7,  0,  0
122:            24, 17, 11,  8,  2, 27,  0,  1,  3,  0,  0
123: /          AL  FB
124:             6,  5)
125:
126: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
127: (@15,      31,  0,  0,  0,  0, 24,  0,  4,  5,  0,  0
128:            22, 18,  1,  8,  1, 12,  0,  4,  5,  0,  0
129:            31,  0,  0,  0,  0, 18,  0,  4,  3,  0,  0
130:            22, 18,  1,  8,  1, 18,  0,  4,  3,  0,  0
131: /          AL  FB
132:             4,  7)
133:
134: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
135: (@20,      31,  0,  0, 15,  0, 23,  0,  4,  0,  0,  0
136:            31, 22, 19, 15,  8,  4,  0,  2,  0,  0,  0
137:            31, 24, 23, 15,  6,  4,  0,  2,  0,  0,  0
138:            31, 20, 21, 15,  6,  5,  1,  1,  0,  3,  0
139: /          AL  FB
140:             5,  7)
141:
142: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
143: (@21,      31,  0,  0, 15,  0, 20,  0,  1,  0,  2,  0
144:            31,  0, 25, 15,  0,  0,  0,  0,  0,  1,  0
145:            31,  0, 26, 15,  0,  0,  0,  1,  0,  1,  0
146:            31, 29, 20, 15, 11,  4,  0,  2,  0,  0,  0
147: /          AL  FB
148:             5,  7)
149:
150: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
151: (@22,      26,  0, 20, 15,  0, 23,  0,  2,  0,  0,  0
152:            26, 21, 20, 15,  9,  8,  0,  1,  3,  0,  0
153:            27, 22, 25, 15, 10,  8,  0,  1,  0,  0,  0
154:            31, 31, 21, 15, 13,  4,  0,  0,  0,  0,  0
155: /          AL  FB
156:             5,  7)
157:
158: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
159: (@23       31,  0,  0, 15,  0,  0,  0, 15,  7,  3,  0
160:            31,  0,  0, 15,  0,127,  0, 15,  0,  3,  0
161:            31,  0,  0, 15,  0,127,  0, 15,  3,  3,  0
162:            22, 19, 19, 15,  1, 12,  0, 15,  3,  3,  0
163: /          AL  FB
164:             2,  7)
165:
166: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
167: (@24       31,  0,  0, 15,  0,  0,  0, 15,  7,  3,  0
168:            31,  0,  0, 15,  0,127,  0, 15,  0,  3,  0
169:            31,  0,  0, 15,  0,127,  0, 15,  3,  3,  0
170:            19, 31, 13, 15,  1, 12,  0, 15,  3,  3,  0
171: /          AL  FB
172:             2,  7)
173:
174: /------------------------------------------------------------
175: / WAVE DATA
176:
177: .wave_form 8,1 = (26,0)
178:
179: /------------------------------------------------------------
180: / TRACK DATA
181:
182: (t1) / めろ : yコマンドに注意
183:     t91
184:     @v0r*6@k-2
185:     s8,3@s6,3@h28,1
186:     |:
187:     @3@v127o4L16@a1
188:     @q1y216,9c8c8d,*10e,*10y216,6
189:     f8.y216,9e,*10dy184,13y216,0q8c4..
190:     y184,11y216,10e-a-<c>fb-<dy184,15
191:     @q1y216,9c8c8d,*10e,*10y216,6
192:     f8.y216,9e,*10dy184,13y216,0q8c4..
193:     @5_4>>{e-a-<ce-a-<ce-a-<ce-c>a-}4.|@a
194:     |:
195:     @5@v127o5L4.@m1
196:     @q3dy248,40@q5{>b<cd}@q3y248,41dy216,5c8>y216,0b4,*42
197:     <@q5gdgdgfcd
198:     @q3ey216,5@q4>g8<c8@q6e8y216,0@q3ey216,5@q4>a8<c8@q5e8
199:     y216,0@q4fec*60@q2d*84_3@q4fe@q3d*60>@q2b4..@m|
200:     @v121y152,20
201:     |:q8c@q2{>g<cd}{gf}{fgf}
202:     @q2e@q1d8@q2c4{ec}d8e4
203:     >a@q2{ab<c}{fecd}2.:|
204:     @q1^1c@q3{cde-}@q1^c^@q3{dfb-}
205:     :|:|
206:     @3@v127o5L16
207:     @q1y216,9c8c8@q2dey216,6
208:     @q1f8.y216,9@q2edy184,13y216,0q8c4..
209:     y184,11y216,10>e-a-<c>fb-<dy184,15
210:     q8y216,9e8e8@q3fgy216,7b-8.y216,9ay216,0q8g*21&_8g*3^
211:     s,3@s,10@h,12@a4y152,22f*78&_f*4^e*146,*98
212:
213: (t2) / ディレイ & さぶ
214:     @v0r*6
215:     s1,1@s3,2@h28,0
216:     |:
217:     @6@v122@k8o4L16@q1c*18&
218:     c8c8def8.edc4..q8e-a-<c>fb-<d@q1
219:     c8c8def8.edc*66
220:     @8@v119@k7>>{e-a-<ce-a-<ce-a-<ce-c>a-}4.|
221:     |:
222:     @6@v124@k9o5L4.d*18&
223:     @q3d@q6{>b<cd}@q3dc8@q6>b4
224:     <@q5gdgdgfcd
225:     q8e*66@11@q4{>g<ce}q8e@q4{>a<ce}r*6@6
226:     fec*60@q20d*84_3@q4fe@q3d*60>@q2b*66@k0|
227:     @7@v119@k-5o5@c10@m12@a2
228:     |:@q1c@q2{>g<cd}{gf}{fgf}
229:     q8e@q1d8@q2c4{ec}d8e4
230:     >a{ab<c}{fecd}2.:|
231:     @mc{>a-b-<c}>a-@a@10@v118q8{fff}
232:     :|:|
233:     @6@v122@k8o5L16@q1c*18&
234:     c8c8def8.edc4..q8>e-a-<c>fb-*6<@q1
235:     @3@v125t90
236:     y217,9c8c8@q3de
237:     y217,6@t-8f8.y217,9@t-2ey217,0@t-3q8d*21&_8d*3^
238:     s,3@s,10@h,12
239:     @a4t91y153,22c*78&_c*4^c*146,*98
240:
241: (t3) / こーど & さぶめろ
242:     @v0r*6L16s1
243:     |:
244:     @10@v124@k-3L4.y242,57
245:     @q2|:o4{ggg}f8.f8e-4..|{e-f}:|
246:     ^1y242,25<c8.&|:5_3c32&:|`r32|
247:     |:
248:     @3@v110o4|:6b:|b-b-<|:4c:|
249:     @9@v98@k-1o5@s4@h4@m-14
250:     |:3@q20{er}q8@v104
251:     |1{>eg<c}:|
252:     |2{>_10e^a<c}:|
253:     |3r*216:|
254:     @5@v125@k1o5@s3@h28@m13
255:     @q4dc>@q3b*60@q2a-4..@m|
256:     @7@v114o5@m-9q8c*14&|:c{>g<cd}{gf}{fgf}
257:     ed8c4{ec}d8e4>a{ab<c}f8.e8.c8.|d8.:|d*22@m
258:     @15@v119@k-3o3c2.@12o4@q1a-^6{b-<df}
259:     :|:|
260:     @v124L4.@q2|:o4{ggg}f8.|f8e-4..{e-f}:|@q3e*38,*33
261:     y146,31y210,6y242,25g*80q8g*146,*98
262:
263: (t4) / こーど
264:     @v0r*6@k-3
265:     |:
266:     @13@v125L4.
267:     @q2|:o5{ccc}>b-8.b-8a-4..|{a-b-}:|
268:     ^2a-4&|:4_2a-32&:|`>|
269:     |:
270:     @10@v122o4|:8d:||:4e-:||:4e:||:4f:||dfa-2.
271:     @15@v122o2L2.@q4|:gggaaa:|gb-L4.
272:     :|
273:     @10@v125o4
274:     y147,25y179,13y211,13y243,24
275:     L16|:6d:||:12f:||:6d:|
276:     :|
277:     @v125L4.@q2|:o5{ccc}>b-8.|b-8a-4..{a-b-}:|a*37,*35
278:     y147,31y219,6y211,6y203,6<c*81,*77c*146,*98
279:
280: (t5) / こーど : 構成はt4と同じ
281:     @v0r*6@k3
282:     |:
283:     @10@v124L4.y244,57
284:     @q2|:o4{eee}d8.d8c4..|{cd}:|
285:     ^1y244,25e-8.&|:5_3e-32&:|`r32|
286:     |:
287:     @10@v122o3|:6b:|b-b-<|:12c:||>b<df2.
288:     @15@v122o2L2.@q4|:13c:|dL4.
289:     :|
290:     @10@v124o3
291:     y148,25y180,13y212,13y244,24
292:     L16|:6b:|<|:6d:||:6a-:||:6f:|
293:     :|
294:     @v124L4.@q2|:o4{eee}d8.|d8c4..{cd}:|@q3c8.
295:     y148,31y212,6y244,25f*82^3q8e*146,*98
296:
```

▶誰かX68000用のグラフィックアクセラレータを作ってくれませんか（スロット使用）。最近，X68000のグラフィックはやはり貧弱だと思うようになりました。

山内　政明(23)愛知県

```
297: (t6) / 矩形波上
298:     @v0r*6L16@1@v124@k0@q2
299:     @c%1000
300:     s,1@s,2@h,3@a1
301:     o3|:12cc|c:|@2@c%1000@q4c&c4.c4.
302:     |:
303:     @2@v124L*36
304:     >|:8f:||:4e:|e-e-e-(e-ge-)
305:     |:8a-:|gg(ggg)4.f+f+(f+f+f+)4.
306:     |:7f:|(fff)|gggg_8g*144~
307:     |:g+2(g+g+e-g+)4g+4.@q5(g+<cd)4.
308:     @q4e*60>e16a4.f+4.f+@q5(f+df+)@q4
309:     f4.(fff)4.f2(ffga)4:|
310:     g+g+g+g+_14@1@q2|:4(b-b-b-)8.~4:|
311:     :|
312:     L16@1@c%1000|:12g:|<_1|:12g:|~
313:     |:45c:|ceg<_2|:30c:|@2@q4c8.c>~3geq8c*228,*180
314:
315: (t7) / 矩形波下 : t6と似てます
316:     @v0r*6L16@1@v119@k0@q2
317:     o2|:12cc|c:|@2@q4c&c4.c4.
318:     |:
319:     @2@v119L*36
320:     >|:8f:||:4e:|e-e-e-(e-ge-)
321:     |:8a-:|gg(ggg)4.f+f+(f+f+f+)4.
322:     |:7f:|(fff)|gggg_8g*144~
323:     |:g+2(g+g+e-g+)4g+4.@q5(g+<cd)4.
324:     @q4e*60>e16a4.f+4.f+@q5(f+df+)@q4
325:     f4.(fff)4.f2(ffga)4:|
326:     g+g+g+g+_14@1@q2|:4(b-b-b-)8.~4:|
327:     :|
328:     L16@1|:12g:|<_1|:12g:|~
329:     |:45c:|ceg<~2|:30c:|@2@q4c8.c>_geq8c*228,*180
330:
331: (t8) / ぽこぽこどらむ : 長いけど，切り貼りでなんとか
332:     @o
333:     @v0r*6L16@v127@k-15
334: /   [A]
335:     |:
336:     @23o6@q4cc                                             /c
337:     @24o6@q1c                                              /o
338:     @23o6@q4cccccccc                                       /c
339:     @24o6@q1c|                                             /o
340:     @23o6@q1(cc)                                           /c
341:     @22o4@q4@k-45|:@o27(d>a)<@o:|@k-15                     /t
342:     @21o3@q1|:3                                            /s1
343:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@o:|   /s2
344:     @23o6@q1(cc)@q4cccc                                    /c
345:     @24o6@q1c                                              /o
346:     :|
347:     @23o6@q1(cc)                                           /c
348:     @21o3@q1|:2                                            /s1
349:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@o:|   /s2
350:     @20o1@q1|:3d*2&(b>d)10:|                               /b
351:     @21o3@q1                                               /s1
352:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@or8   /s2
353:     |:3
354:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@o:|   /s2
355: /   [B]
356:     |:22
357:     @20o1@q1d*2&(b>d)10                                    /b
358:     @23o6@q4cc                                             /c
359:     @21o3@q1                                               /s1
360:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@o     /s2
361:     @23o6@q4c                                              /c
362:     @24o6@q1c                                              /o
363:     :|
364:     @20o1@q1d*2&(b>d)10r8                                  /b
365:     @23o6@q4cr8                                            /c
366:     @24o6@q1c8c8                                           /o
367:     @23o6@q4c                                              /c
368:     @24o6@q1c                                              /o
369: /   [C]
370:     _1|:
371:     |:3
372:     @20o1@q1d*2&(b>d)10r8                                  /b
373:     @23o6@q4cr8cr8cc                                       /c
374:     @24o6@q1c                                              /o
375:     @20o1@q1d*2&(b>d)10r8                                  /b
376:     @23o6@q4cr8cr8|cr8                                     /c
377:     :|
378:     @21o3@q1y127,14|:3                                     /s1
379:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@o:|   /s2
380:     :|~
381:     @20o1@q1d*2&(b>d)10                                    /b
382:     @23o6@q1_3(cc)c~                                       /c
383:     @21o3@q1y127,14                                        /s1
384:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@o     /s2
385:     @23o6@q4cc                                             /c
386:     @20o1@q1d*2&(b>d)10                                    /b
387:     @23o6@q1cc                                             /c
388:     @21o3@q1                                               /s1
389:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@o     /s2
390:     @23o6@q4c                                              /c
391:     @24o6@q1c                                              /o
392:     @20o1@q1d*2&(b>d)10                                    /b
393:     @23o6@q4_3cc~ccc                                       /c
394:     @22o3_10y127,0
395:     |:3@o14c*4&@o11c*3&@o9c*2&@o7c*2&@o4c*1`@o:|~
396:     @22o4@q4@k-45@o27_3(d>a)6<~(d>a)6@k-15
397:     @21o3@q1|:2                                            /s1
398:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@o:|   /s2
399: /   [B']
400:     |:24
401:     @20o1@q1d*2&(b>d)10                                    /b
402:     @23o6@q4cc|                                            /c
403:     @21o3@q1                                               /s1
404:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@o     /s2
405:     @23o6@q4c                                              /c
406:     @24o6@q1c                                              /o
407:     :|r*6
408:     @22o4@q4@k-45@o27(d>a)6@k-15                           /t
409:     @21o3@q1|:2                                            /s1
410:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@o:|   /s2
411: /   [A']
412:     |:3
413:     @21o3@q1|:6                                            /s1
414:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@or8:|/s2
415:     @22o3@q4@o20                                           /t
416:     @v115y127,10
417:     |:(b,f)6~4:|<
418:     @v115@o27@k-45
419:     y127,30(d>a)6<~
420:     y127,20(d>a)6<~
421:     y127,10(d>a)6<~
422:     y127,0|:(d>a)6:|@k-15
423:     @21o3@q1y127,8|:3                                      /s1
424:     @o31b*2&@o29b*2&@o24b*2@o:|r                           /s2
425:     :|
426:     @21o3@q1|:3                                            /s1
427:     @o31b*2&@o29b*2&@o24b*2&@o20b*2&@o16b*2&@o12b*2@o     /s2
428:     |r8:|
429:     @22o3@q4@o27_24@o22y127,7(af)@o17(fc)~
430:     @21o3@q1y127,4|:                                       /s1
431:     @o31b*2&@o29b*2&@o24b*2@o:|                            /s2
432:     @20o2q8@b700,-8000,0y215,17>_2bg~dr*216@o              /b
433:
434: (p)
```

リスト4 GRADIUS2のカウンタ表示

```
1:00001C7A 00000000   2:00001C7A 00000000   3:00001C7A 00000000   4:00001C7A 00000000
5:00001C7A 00000000   6:00001C7A 00000000   7:00001C7A 00000000   8:00001C7A 00000000
```

リスト5 NATURE

```
 1: .COMMENT  ORIGINAL [ NATURE ] Composed & Arranged by FUKUCHANG
 2:
 3: /::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 4:
 5: /  ORIGINAL  [[  NATURE  ]]
 6:
 7: /  Programed & Composed & Arranged  by YUUKI FUKUI
 8:
 9: /::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
10:
11: / FOR ZMUSIC.X & PCM8.X
12: / MIDI MODULE : CM-64
13:
14: /----------------------------------------------------------------
15: / TRACK SETUP
16:
17: (I)
18:
19: / CM-64
20:
21: (M17,4000)(AMIDI 2,17)
22: (M18,4000)(AMIDI 3,18)
23: (M19,4000)(AMIDI 4,19)
24: (M20,4000)(AMIDI 5,20)
25: (M21,4000)(AMIDI 6,21)
26: (M22,4000)(AMIDI 7,22)
27: (M23,4000)(AMIDI 8,23)
28: (M24,4000)(AMIDI 9,24)
29: (M25,4000)(AMIDI10,25)
30: (M26,4000)(AMIDI11,26)
31: (M27,4000)(AMIDI12,27)
32: (M28,4000)(AMIDI13,28)
33: (M29,4000)(AMIDI14,29)
34: (M30,4000)(AMIDI15,30)
35: (M31,4000)(AMIDI16,31)
36:
37: /----------------------------------------------------------------
38: / CM-64 INIT
39:
40: .ROLAND_EXCLUSIVE 16,22={$7F,00,00,00}
41:
42: /----------------------------------------------------------------
43: / CM-64 SYSTEM SETUP
44:
45: / LA SOUND PART
46: .ROLAND_EXCLUSIVE 16,22 ={
47:       $10,0,0                /ADDRESS
48:       64                     /MASTER TUNE
49:       1,3,6                  /REVERB
50:       4,4,4,4,4,4,4,0,4      /PTL RESERVE
51:       1,2,3,4,5,6,7,8,9}     /MIDI CH#
52:
```

```
53: / PCM SOUND PART
54: .ROLAND_EXCLUSIVE 16,22 ={
55:         $52,0,0                 /ADDRESS
56:         64                      /MASTER TUNE
57:         1,4,7                   /REVERB
58:         7,7,7,7,0,0             /PTL RESERVE
59:         10,11,12,13,14,15}      /MIDI CH#
60:
61: /--------------------------------------------------------------
62: / MML DATA SET
63:
64: (T26)       T180
65: (T26)       @38O4L8@V90@P94@K-1
66: (T26)       [DO]
67: (T26)       CGA+<C>A+GA+G<C>FA+G<DCDC>
68: (T26)       [LOOP]
69:
70: (T29)       @38O4L8@V90@P34@K1
71: (T29)       [DO]
72: (T29)       CGA+<C>A+GA+G<C>FA+G<DCDC>
73: (T29)       [LOOP]
74:
75: (T27)       @38O4L8@V90@U100R16.@K0
76: (T27)       R1R1R1R1
77: (T27)       [DO]
78: (T27)       CGA+<C>A+GA+G<C>FA+G<DCDC>
79: (T27)       [LOOP]
80:
81: (T28)       @35O4L8@V86@U100@P34
82: (T28)       R1R1R1R1R1R1R1R1
83: (T28)       [DO]
84: (T28)       CGA+<C>A+GA+G<C>FA+G<DCDC>
85: (T28)       [LOOP]
86:
87: (T30)       @30O4L1@V60@P94
88: (T30)       R1R1R1R1R1R1R1R1R1R1R1R1
89: (T30)       [DO]
90: (T30)       '>G<CE'&'>G<CE'&
91: (T30)       [LOOP]
92:
93: (T31)       @35O4L8@V86@U100@K-1
94: (T31)       R1R1R1R1R1R1R1R1
95: (T31)       R1R1R1R1R1R1R1R1
96: (T31)       [DO]
97: (T31)       CGA+<C>A+GA+G<C>FA+G<DCDC>
98: (T31)       C<C>A+GA+G<C>FA+G<DCDC>
99: (T31)       [LOOP]
100:
101: (T17)      @6O3L1@V98
102: (T17)      |:4R1R1R1R1R1R1R1R1:|
103: (T17)      [DO]
104: (T17)      C2.C1&C4
105: (T17)      [LOOP]
106:
107: (T25)      @121O5L1@V127
108: (T25)      |:6R1R1R1R1R1R1R1R1:|
109: (T25)      [DO]
110: (T25)      |:3R1R2D+4D+4:|
111: (T25)      R1R2D+4D+8D+8
112: (T25)      [LOOP]
113:
114: (T18)      @110O5L4@V122@P34@K1
115: (T18)      |:8R1R1R1R1R1R1R1R1:|
116: (T18)      [DO]
117: (T18)      <C1&C2>A+4<C2.>A+4<C2D2E4F1&F4E2D1&D1&D4>
118: (T18)      <C1&C2>A+4<C2.>A+4<C2D2E4>A+1&A+1&A+1&A+1
119: (T18)      <C1&C2>A+4<C2.>A+4<C2D2E4F1&F4E2D1&D1&D4>
120: (T18)      <F1&F2E4D2.C4D2>A+2<C1&C1&C1&C1&C4>
121: (T18)      |:R1R1R1R1R1R1R1R1:|
122: (T18)      <G1&G2F4E2.D4E2C2D1&D1&D1&D1&D4>
123: (T18)      <F1&F2E4D2.C4D2>A+2<E1&E1&E1&E1&E4>
124: (T18)      <G1&G2F4E2.D4E2C2D1&D2.E4F1&F1&F4>
125: (T18)      <F1&F2E4F2.E4F2D2E1&E1&E1&E1&E4>
126: (T18)      <G1&G2F4E2.D4E2C2D1&D2.E4F1&F1&F4>
127: (T18)      <F1&F2E4D2.C4D2>A+2<C1&C1&C1&C1&C4>
128: (T18)      |:R1R1R1R1R1R1R1R1:|
129: (T18)      <C1&C2>A+4<C2.>A+4<C2D2E4F1&F4E2D1&D1&D4>
130: (T18)      <C1&C2>A+4<C2.>A+4<C2D2E4>A+1&A+1&A+1&A+1
131: (T18)      <C1&C2>A+4<C2.>A+4<C2D2E4F1&F4E2D1&D1&D4>
132: (T18)      <C1&C2>A+4<C2.>A+4<C2D2E4>A+1&A+1&A+1&A+1
133: (T18)      <C1&C2>A+4<C2.>A+4<C2D2E4F1&F4E2D1&D1&D4>
134: (T18)      <F1&F2E4D2.C4D2>A+2<C1&C1&C1&C1&C4>
135: (T18)      |:R1R1R1R1R1R1R1R1:|
136: (T18)      [LOOP]
137:
138: (T19)      @110O5L4@V122@P94@K-1
139: (T19)      |:8R1R1R1R1R1R1R1R1:|
140: (T19)      [DO]
141: (T19)      <C1&C2>A+4<C2.>A+4<C2D2E4F1&F4E2D1&D1&D4>
142: (T19)      <C1&C2>A+4<C2.>A+4<C2D2E4>A+1&A+1&A+1&A+1
143: (T19)      <C1&C2>A+4<C2.>A+4<C2D2E4F1&F4E2D1&D1&D4>
144: (T19)      <F1&F2E4D2.C4D2>A+2<C1&C1&C1&C1&C4>
145: (T19)      |:R1R1R1R1R1R1R1R1:|
146: (T19)      <G1&G2F4E2.D4E2C2D1&D1&D1&D1&D4>
147: (T19)      <F1&F2E4D2.C4D2>A+2<E1&E1&E1&E1&E4>
148: (T19)      <G1&G2F4E2.D4E2C2D1&D2.E4F1&F1&F4>
149: (T19)      <F1&F2E4F2.E4F2D2E1&E1&E1&E1&E4>
150: (T19)      <G1&G2F4E2.D4E2C2D1&D2.E4F1&F1&F4>
151: (T19)      <F1&F2E4D2.C4D2>A+2<C1&C1&C1&C1&C4>
152: (T19)      |:R1R1R1R1R1R1R1R1:|
153: (T19)      <G1&G2F4E2.D4E2C2D1&D1&D1&D1&D4>
154: (T19)      <F1&F2E4D2.C4D2>A+2<E1&E1&E1&E1&E4>
155: (T19)      <G1&G2F4E2.D4E2C2D1&D2.E4F1&F1&F4>
156: (T19)      <F1&F2E4F2.E4F2D2E1&E1&E1&E1&E4>
157: (T19)      <G1&G2F4E2.D4E2C2D1&D2.E4F1&F1&F4>
158: (T19)      <F1&F2E4D2.C4D2>A+2<C1&C1&C1&C1&C4>
159: (T19)      |:R1R1R1R1R1R1R1R1:|
160: (T19)      [LOOP]
161:
162: (P)
```

リスト6 NATUREのカウンタ表示

```
17:00001800 00000180  18:00003000 00008400  19:00003000 00008400  25:00002400 00000600
26:00000000 00000180  27:00000312 00000180  28:00000600 00000180  29:00000000 00000180
30:00000900 00000180  31:00000C00 000002D0
```

(善)の「勝負はこれからだ」

SC-88の話

ローランドの新音源SC-88。いまもって，まだ品薄状態とか。この人気機種をちょっといじる機会があったので，初代SC-55の頃の曲をいろいろと演奏させてみた。

音色をエクスクルーシブなどでエディットしている曲はSC-55とはかなりイメージの違った演奏になるようだ。どうも，音色パラメータの数値と機能の反映のされ方の相関がSC-55とSC-88ではかなり違うようなのだ。たとえば，Oh!X1994年4月号のLIVE inに掲載された早坂真君の「宇宙戦艦ヤマト」のワウワウギターの音。この音はTVFのレゾナンスやCUT OFFフリーケンシーなどのパラメータを書き換えて作り出している音である。が，同じパラメータを設定してもSC-55とSC-88では聴こえ方がずいぶんと違う。パラメータの値とその効果が2つの機種の間で差があるためだ。SC-55ではカッコイイ演奏も，SC-88ではなんだかカエルの鳴き声のように聞こえてしまう。

また，奨励音域をわざと外してトリッキーな音色の使い方をしている曲も同様にイメージの違う演奏になってしまう。SC-88では同名の音色もどうやら新しくサンプリングした波形に置き換わっているようだ。

結局，DTMの達人のデータは「規格」に対応したデータよりもその音源固有の性能をフルに使いきるようなデータとなってしまっているようで，後継機種で演奏した場合はいろいろと問題が出てきている。

音源の性能を上げればデータの互換が危うくなる……，なんかパソコンとソフトの関係に似ているよねぇ。

ドラえもんミュージカルの話

8月13日に，東京の中野サンプラザで公開されたミュージカル「ドラえもん/のび太の恐竜」に行った。ドラえもんはもちろん，のび太やスネ夫までもが「着ぐるみ」で，子供とは思えない大きい図体。大人が黒服を着ただけの悪役・恐竜ハンターたちに囲まれても，のび太たちのほうが体が大きくて強そうなのがおかしかった。そして，舞台上のドラえもんはどんなピンチに陥っても顔が笑っているのが，ある意味無気味だった。

ラジオの話

ある日，車を運転していてふと耳に入ったラジオCM。みんな知っている，北風と太陽のどちらが優れているかを旅人のコートを脱がせることで勝負するおとぎ話がモデルになっている。

しかし，なんとこのCMでは，太陽がジリジリと照らしつけるとその旅人は冷房の効いた喫茶店に入ってしまったというのだ。そこで北風と太陽は引き分け，次に月が現れて美しい女に変身，そしてチュっとキスをすると旅人はコートどころかパンツまで脱ぎ始め，この勝負は飛び入りしてきた月の勝ち……。さて，このような「柔軟な発想」はビジネス戦略を生み出すうえでとても大事，さぁ，あなたも××ビジネス専門学校へ……という内容だ。あまりにも強引な展開と低俗な内容に，この学校，いきなり経営不振に陥ってしまうのではないかと心配になってしまった。誰か，このビジネス専門学校の詳しい情報提供を求む。　(西川善司)

(善)のゲームミュージックでバビンチョ

西川善司

行ってきたぞ ゲームミュージックフェスティバル

7月30日と7月31日の2日間にわたって東京の日本青年館で行われたゲームミュージックフェスティバル，今年も行ってきました。行ってびっくり。なんと全席完売，プレス用の席はないとのこと。ガーン。「雑誌関係者はきっといい席で見てるんだろーなー」と考えている君，実際はそうではない。券を売ってなんぼの世界，タダ見の衆には厳しいのだ。というわけで，最後列の壁によりかかっての鑑賞となってしまった。周りには同じ境遇の各雑誌の関係者がチラホラ。

矩形波倶楽部のメンバーがPOP'Nツインビーの曲に乗って颯爽と登場。コナミお得意のキャッチーメロディは相変わらずいいなぁと酔いしれていると，続いて古川もとあきが登場。古川もとあきといえばギターの名手，っつーことで，XEXEXのBREEZEなどのギターがカッコよく決まる名インスト曲が演奏される。うーん，矩形波倶楽部とはこの音だぁ。ギターの音色に感動しクラクラしていると突然キンキンボイスの乱入。そう，國府田マリ子だ。30日は國府田マリ子お目当てのファンも多かったらしく会場は一時歓声の嵐，騒然となる。もう教祖さまって感じ。このあとは一連のツインビーパラダイス関連のボーカル曲のオンパレード。

続いて登場したのはアルフライラ。メンバー半分を一新しての登場だ。うーむ，なんかレコード会社側の思惑がいろいろと交錯してるんだろーな。新ボーカリストはなかなかうまかったです。ストライダー飛竜のバイオリンアレンジインストもなかなか渋かった。劇場用のサントラ曲やボーカル曲の初披露も試みとしてはよかったが，客の反応はいまいち。そりゃそうだ，当日会場にやってきたのはゲームミュージックファンだもんな。

うーむ，アルフライラはどーもカプコンのサウンドスタッフではないような気がしてきたぞ。だってゲーム会社が「歌しか歌えない女の子をサウンドスタッフにおいておく」わけないし。SSTも結局，本当のセガサウンドチームだったのは2人であとはみんなレコード会社に所属しているミュージシャンだったし，最近の「ゲームミュージック」はずいぶん「商業」の手が入り込んできている気がする。うーむ，喜ぶべきか悲しむべきか，アルフライラのステージを見ていろいろ考えさせられてしまった。

初日のトリを務めたのが葉山宏治&ブラザーズ。会場の半分以上がこれを目当てに来てたみたい。ステージ開幕前から絶え間ない「兄貴」コールが会場を埋めつくし，なにかの宗教儀式のようだった。「兄貴と私」など，観客と一体化してしまうことを意識して作られたライブステージ向きの曲が多いのが印象的だった。客の盛り上げ方は，当日出演バンド中トップであったことはほぼ間違いない。会場では「葉山ファッションを君にも」っていうことで「超兄貴バンダナ」なども売られていた。

以上が初日のレポート。翌31日は大御所ZUNTATA，そして何でもありのGAMADELIC，SNK新世界楽曲雑技団が登場した。SNK新世界楽曲雑技団はゲストに千葉麗子なども登場し，必然的に会場にはゲームミュージックファン以外のアレ系の人の姿も。SNK新世界楽曲雑技団はSNKサウンドチームの人間は皆無で，さっきいったいわゆる商業的に結成された即席バンドなのだが，2日間を通していちばん原曲に近いアレンジで，ファンの期待を裏切らない素晴らしいものだった。

というわけで来年はどーなるか。今年不参加だったJDKやSST→B-Univ.は参加するのか。来年は雑誌関係者にも席があるのか。いまからいろいろと楽しみだ。

＊　＊　＊

●NEO・GEO SUPER LIVE! 1994
SNK新世界楽曲雑技団
CD：PCCB-00161　2,500円(税込)
ポニーキャニオン　10/5発売

7月31日のゲームミュージックフェスティバルでのSNK新世界楽曲雑技団のステージを収録したライブアルバム。レコード会社の都合かどうかは知らないが，千葉麗子が出演している部分はごっそり削除してあるようだ。

お勧め度　7

●ナムコ　ゲームサウンド　エクスプレス Vol.14　リッジレーサー2
CD：VICL-15031　1,500円(税込)
ビクターエンタテインメント　9/21発売

リッジレーサーは「2」の発表に伴ってサウンドのほうもマイナーチェンジ。前作同様シーケンスド・リズムにサンプリングネタをトリッキーに展開したいわゆるテクノハウス系。この手の曲はレースゲームには合わないでしょーなんていう第一印象だったんだけれど，どうしてどうして，車の中で聴いてるとアクセルを踏み込む足になぜか力が入ります。また，これといった旋律もないリズム主体の曲なのでエンジン音との相性もいい。このCD，ドライブに最適の1枚といい切ってもいい……が，くれぐれも事故にだけは気をつけましょう。それとこれを聴きながらの受験勉強はやめましょう。

お勧め度　7

●KAISER KNUCKLE
TAITO ZUNTATA

CD：PCCB-00159　2,000円(税込)
ポニーキャニオン　9/21発売

出た！　タイトーからも「ストII」もどきが。K社のマーシャル某の失敗例もあるし，今後のこのテのゲームの行方は野次馬としては実に興味深いところ。パーツはもろテリー&ガイルじゃねーかーという街のツッコミはすでにタイトーの耳に入っているかしらん。さて曲のほうはナショナリティ系音楽に固執しない，タイトーらしいサウンドが満載。キャラクターボイスはアニメ界で著名な声優を起用しているので，CD末尾のSE集ではテレビで耳慣れた声が飛び出してくるぞ。

お勧め度　8

●龍虎の拳2
VHS：PCVP-11437　4,800円(税込)
ポニーキャニオン　発売中

ゲームキャラクターの声優がおしゃべりを繰り広げながらゲームを紹介していく，プローモーション系の内容。とはいってもさりげなく対CPU戦のCPUのハメ方が紹介してあり，NEO･GEO版でどうしても最後の隠れ挑戦者ギースに会えない人にとってはかなりありがたい情報になるはず。それにしても声優さんはひとり3役くらいをこなしているようで，エンディングクレジットのおしゃべりは誰が何をしゃべっているのかわけがわからなくて楽しい(?)。

お勧め度　6

●スーパーストリートファイターII X
VHS：GV-009　7,000円(税込)
新声社　発売中

他社から発売されるゲームビデオはなんとなくゲームのプロモーション的な内容でイマイチ……と思っている人は少なくないと思う。ゲーメストの新声社から発売されたこの「ストII X」のビデオはそういったものと違って「攻略」に徹底している。対CPU戦はCPUをパターンへ追い込む方法などが惜し気もなく収録されている。さらに対人戦，対CPU戦両方に有効な連続技をなんと総計100以上も紹介。それも実用性の高いものばかり。こんな戦い方があったのかと，思わず目から鱗が落ちるシーンが至るところに。ビデオ後半はストII X全国大会の模様がダイジェストで収録されている。こちらも見応えあり。ストII Xファンは必見。

お勧め度　9

●ワイルドトラックス
CD：SRCL2960　2,800円(税込)
ソニー・ミュージックエンタテインメント　9/21発売

さすがソニー。アレンジバージョンが相変わらず強力だ。トラック1のロック調のタイトル曲からサンバ調のAQUA TUNNELへのノンストップ構成はお見事。2トラックのリズミカルなウェスタンも聴いているだけで元気が出てきそう。ここまでいいと「ロックマンX」のときのようにアレンジバージョンアルバムも出してほしかった気もする。オリジナル曲は全曲収録，CD最後にはSE集も。

お勧め度　9

●究極戦隊ダダンダーン
おそるべき悪魔の陰謀
CD：KICA-1151　2,800円(税込)
キングレコード　9/21発売

キングレコードお得意のドラマCD。昔懐かしタイムボカンシリーズのノリをゲームに盛り込んだダダンダーンということで，ドラマCDのほうもその筋の声優を起用して世界観をよりリアルに浮かび上がらせることに成功している。ただ，ドロンジョやトンズラーの声優は本家の人ではないようでこのあたりの不徹底さが気になった。まぁ，ボヤッキーの「ポチっとな」が聴けたからよかったけど。ナレーションも解説マンの富山敬だしね。このCD，ゲームファンはもちろん，タイムボカンシリーズファンの人もどーぞ。主題歌は今回も子門真人が熱唱してます。

お勧め度　8

●ぽっぷるメイル　パラダイス
CD：KICA-1150　3,000円(税込)
キングレコード　9/21発売

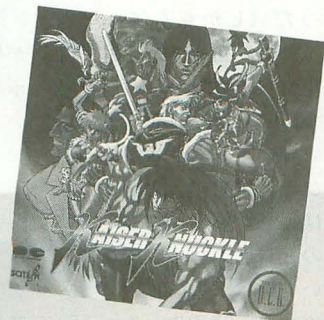
スーパーストリートファイターII X

キングレコードは最近，ドラマCDづくめ。ツインビーが終わったら今度はファルコムの人気ARPG「ぽっぷるメイル」をドラマ化。主役のメイルの「らんま1/2」の林原めぐみをはじめ著名声優が大勢登場する。もうキングレコードのドラマものの顔ともいえる存在になった國府田マリ子も登場。

お勧め度　7

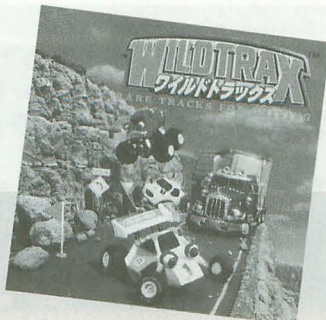
●ストリートファイターII
オリジナルサウンドトラック
CD：SRCL2937　2,800円(税込)
ソニー・ミュージックエンタテインメント　発売中

映画「ストII」のサントラアルバム。映画のBGMをすべて収録，篠原涼子の歌う主題歌，アルフライラの歌う挿入歌も収録。さらにボーナストラックとして，ストII Xの隠れキャラ・豪鬼のオリジナルサウンドも収録している。私も須藤君とこの映画を見に行きました。ところでこの映画のラストシーン，リュウがベガの足を抑え，抱えて「いまだー！　ケーン」「おぉ！食らえベガ!!　昇龍拳!!」……これっていわゆるヒーローモノでは「卑怯者」のやることなんじゃ……？

お勧め度　7

KAISER KNUCKLE

ワイルドトラックス

究極戦隊ダダンダーン

新製品紹介

F-Card V5 for x68k

▼

データベースで快適情報生活

Kiyose Eisuke 清瀬 栄介

実用ソフトのなかでは比較的個人でのニーズも高いのがデータベースであるが，このほど，クレストのX68000用「F-Card」のバージョンアップ版が発売された。

日頃，X68000に実用ソフトが少ないことを嘆いているユーザーは多いらしい。確かに「パーソナルワークステーション」という割には，住所録の作成にもけっこう困ってしまうような状況では，家でのX68000の肩身もちょっと狭いかもしれない。

かといって，家で使うのにビジネス用のヘビーデューティなソフトがあっても，もてあましてしまうのも確か。だいいち値段が高いしね。

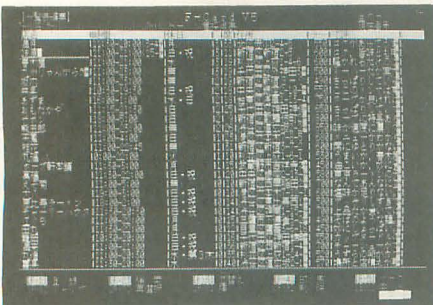
そんな状況に登場したのがこの「F-Card V5」。カード型データベースで，データの管理・集計，グラフの作成などが手軽に行える。しかも12,800円という，PC-9801の世界でも十分バーゲンプライスと呼べる値段だ。

そしてこのF-Card V5，ただひととおりのことができるだけではなくて，プラットホーム機能という，なかなかユニークな機能を持っている。家で使って面白い，こいつはまさにパーソナルワークステーション向けのデータベースなのだ。

▼

プログラム基地になるのだ

▲

カード型データベースというものについては特に詳しく説明することはないと思う。図書館の貸し出しカードみたいに，いくつかの項目の入ったカードを作って，それを参照するというのがカード型データベースである。項目についても種類があり，「数字」なら集計ができるとか，文字なら「検索」ができるというわけだ。

で，項目の種類は，普通なら「文字」と「数字」と「選択肢」ぐらいのものなのだが，このF-Card V5には独自の種類がある。「File」と「Prog」というヤツだ。

「File」という種類の項目を作る。そこにファイル名を書いておくと，カードを見ているときにリターンキーを押してファイルの中身を参照することができるのだ。たとえば，

9月原稿　：F-Card V5 レビュー
行数　　　：○行
原稿ファイル：GENKOU3.TXT

というカードを作ると，「原稿ファイル」の欄からいつでも原稿の中身をタイプできるというわけ。

同じように「Prog」ではなんとプログラムの実行ができてしまう（もちろんオプションも記述することができる）。他プログラムとリンクしたカードを作れる，これがF-Cradのプラットホーム機能である。

この機能は僕の知っているデータベースにはなかったので，初めて知ったときには，新鮮だった。カードの内容が打ち込んだデータだけじゃなくて，独立したプログラムまで取り込んでしまうというのはなかなか斬新である。

それになんといっても面白そう。いろいろ応用例を考えてみると楽しい。たとえばボクのようなライターだったら，自分の書いた文章を「FILE」として入力しておいて，内容や分量に応じて過去の原稿をピックアップできる。普通は，テキストファイルを整理しようと思ったら，ファイル名を工夫したりディレクトリに入れたりする程度なのだが，この機能を使えば原稿の反省点とか担当者までセットにして呼び出せてしまう。

OPMファイルやPICファイルを貯蔵している人などには，すごく使い道がありそうだ。ジャンル・評価・作者などを一緒に入力しておけば，その内容に応じていつでもファイルをピックアップし，表示させたり演奏させたりすることができてしまう。しかも作者のコメントやマニュアルなどのドキュメントもカードデータを通じてリンクさせておける。一歩進んだファイル管理ソフトとしても使えるわけだ。さぁ，どうだ。

▼

独自の機能はほかにもいっぱい

▲

ほかにもこのF-Card独自の機能はある。まず数字読み上げ機能。「数値」に指定した項目については，カーソルを合わせてF・10キーを押すことで，PCMに数字を読み上げさせることができる。

もとからついている声はなかなか威勢のいい女の人の声なのだが，そこはそれ，ファイル名まで公開されているから，好きに変えてしまえばいいのだ。

たとえば「好きなストIIの必殺技」という項目をわざと数値型で作っておく。「1」と記入されているところでF・10キーを押すと「しょーりゅーけん！」と発音するとか……。

ウーン，これはイマイチつまらなかったが，とにかく発想次第でいろいろ使い道はありそうだ（しかもくだらない使い道が……)。この，「遊べる」ということが個人向け実用ソフトではけっこう大事なポイントなので，F-Card V5はなかなかいいところをついている。

もうひとつは自由入力画面。

普通はカード型データベースというのは，その名のとおり，カード型の構造をしている。たいていは入力画面も「カード型なんだからカードっぽくやっちゃえ」とデザインされているのだが，このF-Cardでは画

一覧表形式はデータの比較をするのに便利

通常の入力画面。あとから項目設定を変えることもできる

面を好きなようにレイアウトして，カードの記入をすることができる。

たとえば俳句のデータベースを作っていて，5・7・5がキレイに画面に収まんなきゃヤダとかいうように，決まった書式に従ってカードを参照したいときに，この自由入力画面を使うといい。

ね，見れば見るほどパーソナル向け，個人が使って楽しい機能があるデータベースなのだ。

自由書式編集。ラーメン屋のミシュランチェック例だ

印刷画面をデザイン。縦倍角と毛筆は横のマーキングで指定する方式

操作性をもっと向上してほしい

もちろん，普通のデータベースソフトが備えている機能はF-Card V5にも装備されている。

自由記入画面のほかに，通常のカード1枚ごとに入力していく画面もあるし，一覧表形式で入力することもできる。項目の自動記入機能や，検索，ソートなどもひととおりそろっている。理系の学生のニーズについていけるかどうかは自信がないが，日常使うソフトとしての機能は十分だ。

ただ，弱いところを探すとすれば，やや印刷関係に難があるかもしれない。なにせこのソフト，自由入力画面でレイアウトをデザインできるくせに，それを出力するときには，プリントアウト用のレイアウトをまた別に作らなければいけないのである。

しかも，画面が仕上がりと近いとはお世辞にもいえない。家庭用ワープロでもプレビューを見られるのだから，こういう点についてはもう少し頑張ってもらいたかった。グラフも簡単な作成機能がついているけれど，これも仕上がりがどうなるのかがわかりづらい。

ファンクションキーを使った操作性も一種独特で，慣れを要する。検索や置換のときはメニューを開いて選択するのだが，そのときカーソルのいる場所は現在やっている作業とは違うのだ。うっかり開いてしまうとキャンセルするのが大変だ。

このあたりは個人にウケそうなデータベースだからこそ気をつかってもらいたい。実用ソフトだから，X68000用にしては異常に地味な外見には目をつぶるが，操作性が良くないのは困ってしまう。仕事ならガマンできることも，趣味でやっていると簡単に放りだしてしまう。

毛筆書体の印字サンプル

【ラーメン屋さんの採点簿ナリ】

店の名前 味源

三ツ星評価 3

コメント

コク，旨味とも申し分なし。ギョーザがあれば．．．

営業時間 11:00-21:00(15-17休)

休日 日・祝

データベース生活のススメ

さて。話はかわって，ボクはPC-286を使って原稿を書いている。ASKがどうしても好きになれないからだが，このPC-286のHDに放りこんであるのが「Ninja3」というちょっと古めのカード型データベース。ワープロ機能もついているので，統合型ソフトみたいな感じで使っている。何を隠そうこの原稿もNinja3で書いていたりするのだ（自分でも変なヤツだと思う）。

普通，パソコンユーザーはワープロをパソコンライフの中心ソフトとして位置づけていると思うんだけど，この「カード型データベースをHDに放りこんでおく」というスタイルも，やってみると意外に便利なものだ。

たとえばグルメ本を手に入れて，データベースを作ってみる。店の名前や場所を入れておけば，「明日荻窪に出かけるけど，どんな店があったっけ」と検索すれば情報がズラリと出てくる。遅くまでやってる店を調べたいとか，あそこの店の休日はいつだったっけとか，いろいろ知りたいことに合った情報の見方ができるのだ。で，行って来たら寸評でもつけておけばオリジナルのミシュランチェックになるというわけ。

ほかにもCDの購入記録や，レンタルビデオを借りてきて自分の映画評をファイリングするとか。PDSやミュージックデータの管理をするのでもいいし，ゲームやシナリオのアイデアを思いついたときに入れておいて，使いたいときに条件に合わせてピックアップするなんてのもいい。

データベースって，何か使うきっかけがないと買わないたぐいのソフトだけれど，思い切って購入すると，なかなか楽しみは多い。とにかく「あとから何度もアクセスするような情報をファイルするなら，カード型データベース」だ。

そんなわけで個人が趣味で使うデータベースとなると，実用的であることはもちろん，入手した情報をどう「面白く」いじれるかということがポイントになる。

そういう意味では，このF-Card V5はなかなか面白いソフトだといえそうだ。ハードディスクにインストールして，「エコロジー」やDIのような存在として使ってみたい1本である。

今後の期待としては，ボクはクレストさんにSX-WINDOW上でF-Cardのような面白いコンセプトをもったデータベースを作ってほしい。ウィンドウアプリケーションなら，操作性も一定の範囲に落ち着くし，データのやりとりも楽だし，いっそう面白いことができそうなんだけど，どう？

X68000用 3.5/5"2HD版 12,800円(税別)
クレスト ☎03(3418)5993

"実戦!"ゲーム作りの

KNOW HOW

〈応用編その1〉

BG実画面拡張の試み

◆

Taguchi Atsushi 田口 敦

今回は，BGの実画面拡張システムを作ります。これによって，制限のあるBGの実画面領域をマップデータに基づいて，実画面以上の大きさを扱えるようになります。1994年3月号も併せて読むと，より理解が深まるでしょう。

最近，第2次世界大戦のミリタリーに興味をもっています。日本の海軍などは世界最強と呼ばれる兵器をいくつももっていました。戦艦大和や零戦などは，戦争に興味がなくても一度は聞いたことがあるでしょう。

そんな日本軍が負けた最大の理由を現代ではこう表現しています。“人材資源の不足”と。

当時の日本人は手先が器用だが，遠くを見渡すのが苦手だったといわれています。しかも，一度調子にのると舞い上がったまま身を引き締めず，さらに成長をしようとしなくなるそうです(さすがに戦後，痛い目にあった人はだいぶ変わったそうですが)。

手先の器用さで産んだ最強兵器零戦や大和，初期の連戦連勝で目の前しか見えなくなったあげく，気がついたら大和は無用の長物となっているし，零戦は敵機に太刀打ちできない代物になってしまいました。少なくとも太刀打ちできたかもしれないベテランパイロットは，人材軽視のためにどんどん失われていきました。

戦略性がなく，戦術&戦闘重視の日本軍が前線の人材を失った結果はご存知のとおりです。

なんだか，これって現代にすごく似ているような気がしませんか。兵器というのをコンピュータに当てはめると，日本人お得意の細かい作業で大容量，低価格のメモリを作り，世界中で売れに売れて喜んでいるうちに日本の半導体産業は基本的にメモリしか作れなくなってしまいました。しかもそのメモリもいまやアメリカや台湾に抜き返されています。1つだけ例を出しましたが（パソコンを例に出すと過激発言が多くなってしまうような気が……），これだけではないはずです。ソフトウェアだってそのまま戦略や戦術，戦闘という言葉に置き換えられると思います。通信産業だってその中の大きな1つでしょう。

僕は日本にトップになってほしいといっているのではありません。世の中は競争で成り立っています（共存共栄といったほうが角が立たないかな)。

これを読んでいるあなたは，読者層から察するにどこかの分野の最前線にいるか，前線に出るために出撃準備していると思います。現在の日本も慢性的な人材不足と呼ばれています（人手不足じゃないっすよ)。そして，諸外国にどんどん追い抜かれて負けの一方になっている感じがします。戦後日本を復興させた人たちがいたように，次に日本(世界&宇宙)を盛り上げるのは我々です。

もう，ガンガン前に進むっきゃないっしょ。

今回制作したBG実画面拡張システム

◆3月号のおさらい

脱線はこのくらいにして，さっそく本題に入りましょう。今回はBG画面の応用ということで，BGの実画面の拡張を試みてみたいと思います。

さて，その前に3月号のおさらいをしましょう。

BGの実画面は，通常表示画面の縦2倍横2倍の広さをもっています（変則画面を除く)。スクロールが必要なゲームなどでは移動範囲が収まればよいのですが，ちょっと派手なものを作ろうとすればこれだけでは足りないのが目に見えてます。そこで，マッピング処理という方法を使用し，用意したデータに従ってBGを書き換え，実画面の大きさ以上の画面を使用可能にするわけです。

3月号は簡略版ということで，1次元的（単純な縦横移動）なスクロールによる書き換えの手順などを取り上げました。

簡単なシューティングゲーム（サイド&トップビュータイプ）ならあれで対応できたと思います。

アルゴリズムは簡単で，スクロールに従ってこれから見えると思われる場所を書き換えるだけです。あのときのサンプルは書き換える量が少なかったので帰線期間中に転送を行っていますが，帰線期間中のほかの処理がいっぱいだとかいうときや，書き換える量が多いときは，表示期間中に転送することも考えなければなりません。

◆BGの実画面疑似拡張

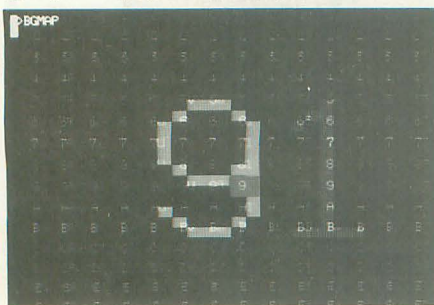
なんだかたいそうなタイトルですが，基本は3月号と同じなので逃げ腰にならずに読み進めてみてください。

ところで，好奇心NoWaitの読者は「実画面の疑似拡張とはいったいなんぞや？」と思われたでしょう。「スクロール処理とどこが違うの？」という素朴な疑問を抱く方もいるかもしれません。いい質問です。ここで，スクロールマッピング処理と疑似拡張処理をはっきり定義してしまいましょう。

●スクロールマッピング処理

ユーザーが処理ルーチンに移動量を引数として渡すプログラム。移動量に応じて書き換えを行う

●実画面疑似拡張処理

ユーザーが処理ルーチンに座標を引数として渡すプログラム。この座標指定範囲は

用意するマップデータに依存する。指定した座標を前回指定された座標と比較し，その差分に応じて書き換える

つまり，スクロールマッピング処理はできることが限定的で扱いにくい反面，処理が単純で高速であるという長所があります。実画面疑似拡張のほうは，用意したマップデータの範囲内での座標指定ができるので，使用方法が非常に簡単で処理プログラムの独立性も前者に比べて非常に高いことが長所に挙げられます。マップデータの範囲内での座標指定というのがわかりにくいと思いますので，説明しましょう。

通常，BGの実画面は64×64キャラクタの大きさをもっていて表示画面が512ドットモードの場合，1024×1024ドットの範囲内を指定することができます。

たとえば512×512キャラクタ分のマップデータを用意して表示画面を512ドットモードにした場合，1つのキャラクタの大きさは16ドットですから，

512キャラ×16ドット＝8192ドット

の範囲を指定することができるようになります。ユーザーはこの範囲内のドットを指定するだけでルーチン側が必要な部分だけ書き換える処理を行います。つまり，ユーザーは用意したマップデータ分だけ実画面が広がったと錯覚できるわけです。

ただ，欠点としてスクロールマッピング処理に比べて処理が複雑になりがちなので，処理の規模に応じて使い分けるべきでしょう。

◆アルゴリズム

今回も依然としてサンプルにできるマップエディタがないので（誰か作ってくれるといいな），グラフィック画面に適当なキャラクターを描いて，それをマップデータとして使用することにします。

まず，書き換えのタイミングの説明をしたいと思います。

3月号は書き換えを割り込み内で行っていました。前述したとおり書き換えの量が少なく，あまり量的な変化がなかったのと，帰線期間を使ってスプライトの表示をOFFにして一気に転送したほうが効率よく処理できたからです。なぜスプライト表示をOFFにすると高速転送できるかは説明しません。わからなかったらX68000のハードウェア本を参照してください。

今回は，転送する量が常時変化するのと，書き換えの範囲が広いので，表示期間中に処理をします。割り込み処理中には処理量が大幅に変化するルーチンや，処理量が多く負荷のかかるルーチンは置かないほうがいいのです。なぜなら割り込み処理が終わる前にまた割り込み（同じ割り込み）が入ったらどうなるかわかるはと思います（わからない人はお勉強するように。結構重要だからね）。

ということで今回の割り込み処理はスクロールレジスタの設定しかやっていません。スクロールレジスタの設定だけは帰線期間中にやらないとちらつきますからね。

では，図1を見てください。これは書き換えと割り込みのタイミングです。見てのとおり通常処理中に指定された範囲のBGを書き換えています。この通常処理中というのは，普通ゲーム中のスプライト管理や当たり判定処理などをやっています。

さて，このBG書き換え処理は，いったんルーチンの内部に入ると帰線期間になろうが，次の同期に突入しようが，終わるまで処理を返しません。通常のスクロール処理ではそんなに処理時間がかかることはありませんが，ほかの処理で負荷がかかっているときに全画面書き換えなどの重い処理を行うことになれば十分考えられます。もし，表示期間中に垂直帰線割り込みがきて，勝手にスクロールレジスタを変更されたら書き換えているところが見えてしまい，とってもみっともないからです。

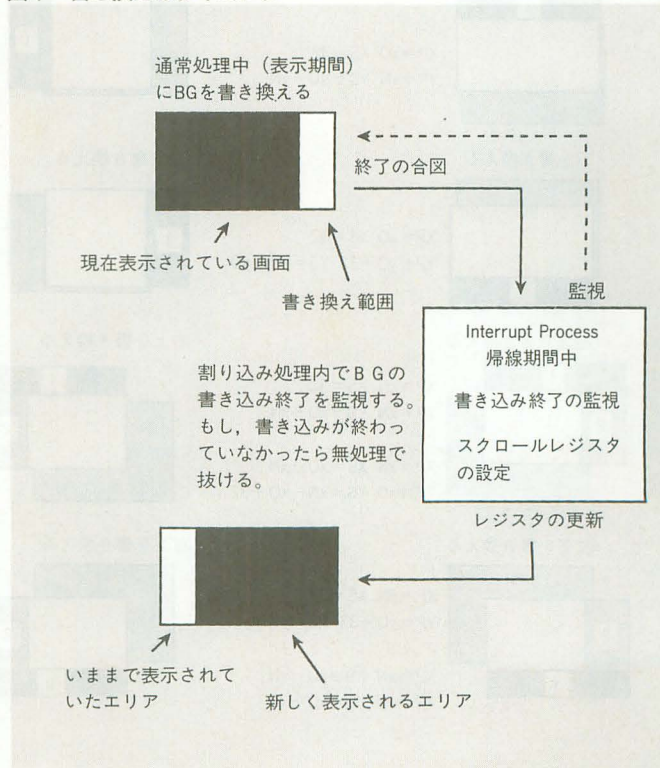
そこで，割り込み処理のほうで書き換えを毎回監視し，書き換えが終わっていなかったらスクロールレジスタを変更しないようにするのです。そうして，書き換え処理が終わったら，レジスタを更新するという処理を行うのです。なぜ割り込みを使ってこんな面倒なことをやるのかというと，見かけ上の処理落ちを軽減するためです。この方法をとれば，もし急激な負荷がかかったり，処理の遅い機種で大量の書き換えを行った場合に1同期で処理が終わらなくてもBGの書き換えに限って2同期内で収まっていればコマ落としのような処理をするのです。まあ，あくまでも軽減するっていう程度ですけどね。

図1　書き換えのタイミング

そして，図2がBGを書き換えるための式と図です。なにやらごちゃごちゃ書き込んでありますが，そう難しく考えることはありません。これはプログラム化するために，ここに示されている図を式にしただけです。本当は1つの数式を出してしまえばいいのですが，変に難しく考えて動作不安定なものより，多少効率が悪くてもわかりやすく確実に動くものを作るのがプログラムの基本(？)なので，とりあえず地道な方法を選んでいます。

図2を見てわかるとおり，各方向に移動した際の書き換える場所とその書き換える範囲の式を示しています。図2の詳細は見ればわかるとして，見方の説明をしましょう。

現在表示されている画面を1，新しく指定された表示されるべき画面を2とします。そして1と2の座標の差分を取ることによってその差分を書き換える範囲として使用します。ただ，その考え方は完全に表示画面外に移動する場合と，斜めに移動する場合は単純に適用されません。

まず完全に表示画面のドット数以上に移動した場合です。この場合はBGの現在表示されていないエリアに全面書き換えします。そしてもう1つ，斜めに移動する場合です。これがなぜ問題なのかというと，差分の式

をX軸とY軸に単純に使用した場合に書き込み範囲が重なってしまう領域ができてしまうのです。それでは非常に効率が悪いので（もしも，512ドットモードのときにXY軸ともに511ドット移動なんてことをやったら，ほぼ同じ画面を2回書き換えることになってしまいますからね），それも考慮してあります。

この式はBGデータエリアのキャラクタ数が縦横ともに64キャラクタであることを前提にしています。X68000の場合256ドットモードだろうが512ドットモードだろうが，ともに64キャラクタなので問題はないでしょう（IOCSで指定できる以外の変則モードを使用すると多少変更する必要が出てきます）。

◆サンプルについて

それでは，さっそくサンプルプログラムの説明をします。今回用意したサンプルプログラムは2本。それぞれ別のプログラムなので，個別にアセンブル&リンクしてください。

まずは，本体のリスト1です。ちょっと長いのですが，同じようなところがあるのでカット&ペーストしてください。

このプログラムは，グラフィックに表示した絵をマッピングデータとして使用し，BGに表示します。グラフィックの初期化を実画面1024ドット，表示画面512ドットに設定したので，このプログラムの場合XYともに，

1024×16＝16384ドット

の実画面拡張が可能となります。

78行あたりまでは各種設定です。レジスタの定数設定は見てのとおりなので問題ないと思います。ただ，47，48行の数値を変えることによってユーザーの用意したマップの大きさに合わせることができます。そのときは，用意するマップの広さが2の倍数の大きさになるようにしてください。

その直後から167行目までは各初期設定です。PCGとパレットの設定をしています。画面初期化後グラフィック用のパレットをスプライトパレットにコピーしています。グラフィックとの相関関係をわかりやすくするために半透明を使用しているので，カラーコードの最下位ビットを立てています。半透明を使ったことのない人は参考にしてみてください。

173行目からの4行はプログラムの表示期間同期合わせです。これは特に説明しません。

ここからが重要です。178行からユーザープロセスが始まっています。ここには自由にユーザーのプログラムを置くことができます。今回はサンプルということで1つのソースに埋め込んでいますが，本来ならば独立性を高めるためにサブルーチン化するか別ファイルにするべきでしょう。ここでやっていることはカーソルキーが押されたらBGの疑似実画面を押された方向に移動することです。0～16384までの数値を疑似スクロールレジスタである，

X_SCRL　及び　Y_SCRL

に書き込むことで移動をさせることができます。それぞれ両方とも16ビットです。

そのほか，表示しているマップを明確にするためにグラフィックを同時にスクロールさせています。中央の色が変わっている部分（スプライト）が，現在表示されているマップです。加速移動については今回も固定小数点を使用しています。この固定小数点の概念は新しいゲーム機（コンシューマ）でも多用されるので，覚えておくと得かもしれませんよ。

そして，232～297行までがこのプログラムのメイン部分です。疑似スクロールレジスタを解析して

図2　BGを書き換える範囲の式

それを適当な処理に振り分けています。振り分け方は先ほどアルゴリズムのところで説明してあるとおりです。

あとはESCキーが押されるまでループをします。そんでもって押されたら終了と。まあこんな感じでしょうか。

垂直帰線割り込み処理については，一応セオリーどおりです。スプライトI/Oにアクセスするときは表示をOFFにするのと，スクロール値を変えるのを帰線期間にやることくらいです。唯一違うのがBGの書き換えが終わっていなかったら処理をしないところです。

347行から始まるサブルーチンは実際にBGを書くためのプロセスです。頭にあるのがジャンプテーブルです。メインルーチンから各サブルーチンにジャンプするのに使われます。ここらへんは3月号でもやったので参考にしてください。BGの書き換えのパラメータ設定は図2で式化したものをそのままコード化しただけです。若干最適化してあるので見にくいかもしれませんが，そこは勘弁ってことで。680行までが各方向に対しての処理です。

そのあとのルーチンが前述の処理で使われているBGの転送ルーチンです。パラメータの指定方法などはソースを参考にしてください。ここではグラフィックのデータをマップデータとしてBGに転送していますが，マップエディタなどで作ったマップを使用することもできます。グラフィック画面はどんなモードでも必ず1ドット1ワード構成になっているので，転送時にワード単位で行っていますが，通常のマップエディタで作ったマップデータは1要素（キャラクタまたはセルといったりします）1バイトだったりするので，それぞれの場合に応じて変えてみてください。766行を変更すればOKです。

リスト2はグラフィックに適当なデータを表示するだけのプログラムです。

打ち込むのが面倒な人は，リスト1を実行する前に適当な16色データを表示してください。16色用のデータ表示プログラムといえば主にMAGとPIがありますが，実画面をクリアしてくれるPIのほうがどちらかといえばキレイです。自分がもっているほうを使ってください。

◆改造の心得

このサンプルの場合，問題になるのは改造より，ゲームへの組み込みでしょうか。

実はこのルーチン，計算をいくらかはしょっているせいか，動作が完璧ではありません。誤差修正を多少しているものの，まだ完璧ではありません。画面がたまに乱れたりするのですが，気になるようでしたら631行のMOVE命令にあるイミディエイトデータを33～63の間で増やしてみてください。ただしあまり増やすとパフォーマンスが低下します。

ゲームなどへの組み込みは，流れを追ってみれば特に問題はないと思いますが，疑似スクロールレジスタへの数値セットは通常処理の一番あとにしたほうがいいでしょう。先頭にもってくると処理落ちの第1原因になる可能性があります。

また，今回は結果を見やすいように16色画面を使用しているのでPCGを16個しか設定していませんが，スプライトエディタで作ったPCGとマップエディタで作ったマップを組み合わせれば実際のゲームか，グラフィックエディタの高速ルーペ，デモやプレゼンテーション用の巨大字幕表示プログラムなどに使えます。

プログラムの説明のところでも述べていますが，マップエディタで作ったマップデータを使用するときはデータのビット幅に気をつけてください。データを用意したら，用意してあるデータのアドレスを，このサンプルで設定されているグラフィックVRAMのアドレスと変えるだけです。

◆次回の当てにならない予告

今回は結構高度(？)で，少々基本からずれたことをやったので次回は基本的かつ応用範囲の広いパレット操作をしたいと思います。フェードイン/アウトやパレットチェンジによる物体の疑似移動効果など，ゲームによく使われているので，利用価値が高いと思います。

それでは今月はこれまで。次回は12月号を目標にしています。

協力　Fill in Cafe' Co.,Ltd.

リスト1　BGMAP.S

```
 1: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
 2: *
 3: *        BGのマッピング処理（データエリアの疑似拡張）
 4: *
 5: *                                                  ESC = 終了
 6: *        16色のグラフィック画面をBGのマッピングデータとして
 7: *        使用します
 8: *
 9: *        X_SCRL .w = 疑似仮想Xスクロールレジスタ
10: *        Y_SCRL .w = 疑似仮想Yスクロールレジスタ
11: *
12: *        MAP_XSIZE = マップデータのXサイズ（2の倍数であること）
13: *        MAP_XSIZE = マップデータのYサイズ（2の倍数であること）
14: *
15: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
16:
17:          .include       iocscall.mac
18:          .include       doscall.mac
19:
20: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
21: *
22: *                定数&マクロ設定
23: *
24: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
25: * 関連I/Oアドレス
26: GVRAM_ADDRES     equu      $c00000 * GVRAM_ADDRES
27: GRPH_PALET       equ       $e82000 * グラフィックパレット
28: SP_PALET         equ       $e82220 * スプライトパレット No.1
29: VDC_R2           equ       $e82500 * ビデオコントローラレジスタ2
30: VDC_R3           equ       $e82600 * ビデオコントローラレジスタ3
31: BG_CTRL          equ       $eb0808 * BGコントローラー
32: PCG_AREA         equ       $eb8000 * PCGエリア
33: BG0_VRAM         equ       $ebc000 * BG0データエリア
34: SSR              equ       $eb0000 * スプライトスクロールレジスタ
35: BSR0_XPOS        equ       $eb0800 * BGスクロールレジスタ0X
36: BSR0_YPOS        equ       $eb0802 * BGスクロールレジスタ0Y
37: GSR0_XPOS        equ       $e80018 * グラフィックスクロールレジスタ0X
38: GSR0_YPOS        equ       $e8001a * グラフィックスクロールレジスタ0Y
39: GSR1_XPOS        equ       $e8001c * グラフィックスクロールレジスタ1X
40: GSR1_YPOS        equ       $e8001e * グラフィックスクロールレジスタ1Y
41: GSR2_XPOS        equ       $e80020 * グラフィックスクロールレジスタ2X
42: GSR2_YPOS        equ       $e80022 * グラフィックスクロールレジスタ2Y
43: GSR3_XPOS        equ       $e80024 * グラフィックスクロールレジスタ3X
44: GSR3_YPOS        equ       $e80026 * グラフィックスクロールレジスタ3Y
45:
46: * 拡張BGの仮想画面の大きさ
47: MAP_XSIZE        set       1024      * マップのXサイズは1024キャラ
48: MAP_YSIZE        set       1024      * マップのYサイズは1024キャラ
49:                            * X_SIZE  1024*16=16384ドット
50:                            * Y_SIZE  1024*16=16384ドット
51: * スプライトのスクリーンモード
52: SCR_MODE         set       512
53: *SCR_MODE        set       256
54:
55: * グラフィック面の実画面の大きさ
56: GRAPH_XSIZE      set       MAP_XSIZE
57: GRAPH_YSIZE      set       MAP_YSIZE
58:
59: * サブルーチン用パラメータのインデックス
60: GR_Xpoint        set       00        * グラフィックGET用X座標
61: GR_Ypoint        set       02        * グラフィックGET用Y座標
62: MV_Xsize         set       04        * 転送するXサイズ-1
63: MV_Ysize         set       06        * 転送するYサイズ-1
64: BG_Xpoint        set       08        * BGPUT用X座標
65: BG_Ypoint        set       10        * BGPUT用Y座標
66:
67: DI       macro
68:          ori.w    #$0700,sr          * 割り込み禁止
69:          .endm
70: EI       macro
71:          andi.w   #$f8ff,sr          * 割り込み許可
72:          .endm
73: SPOFF    macro
74:          move.w   #%00000000_00000000,BG_CTRL
75:          .endm
76: SPON     macro
77:          move.w   #%00000010_00011001,BG_CTRL
78:          .endm
79:
80: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
81: *
82: *                メインプログラム
```

```
 83: *
 84: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
 85: *
 86: *                          イニシャライズ
 87: *
 88: init:
 89:         sub.l   a1,a1
 90:         IOCS    _B_SUPER            * スーパーバイザモードへ移行
 91:
 92:         move.l  #$100,d1            * 512*512 1024*1024 16色
 93:         IOCS    _CRTMOD             * グラフィックを初期化しない
 94:         IOCS    _OS_CUROF           * カーソルオフ
 95:         IOCS    _SP_INIT            * スプライトI/O初期化
 96:         IOCS    _SP_ON              * スプライト表示
 97:
 98:         SPOFF
 99:
100: toumei:         * 半透明設定
101:         * プライオリティ グラフィック > スプライト > テキスト
102:         move.w  #%00_10_00_01_11100100,VDC_R2
103:         *       半透明モード
104:         *           ↓
105:         move.w  #%000_11_001_01111111,VDC_R3
106:         *半透明機能を利用するためにパレットの最下位ビットを立てる
107:         move.w  #16-1,d0
108:         lea.l   GRPH_PALET,a1
109: toumei00:
110:         ori.w   #$0001,(a1)+
111:         dbra    d0,toumei00
112:
113: col_copy:       * グラフィックパレットをスプライトパレットにコピーする
114:         lea.l   GRPH_PALET,a1
115:         lea.l   SP_PALET,a2
116:         move.w  #16-1,d0
117: copy_loop:
118:         move.w  (a1)+,(a2)+
119:         dbra    d0,copy_loop
120:
121: cg_set: * No.0～No.15までPCGをセットする
122:         lea.l   PCG_AREA,a1
123:         clr.l   d2
124:         move.w  #16-1,d1
125: pcgloop00:
126:
127:         move.w  #$80/4-1,d0
128: pcgloop01:
129:         move.l  d2,(a1)+
130:         dbra    d0,pcgloop01
131:         add.l   #$11111111,d2
132:         dbra    d1,pcgloop00
133:
134:         * スプライトデータの転送
135:         lea.l   sp_scrdat,a0
136:         lea.l   SSR,a1
137:         move.w  #4*2*4/4-1,d0   * スプライト4つ分
138: spatr_set:
139:         move.l  (a0)+,(a1)+
140:         dbra    d0,spatr_set
141:
142:         * BGの初期表示設定をする
143: map_data_set:   * マッピングルーチンに引数を渡す
144:         lea.l   GRAPHtoBG_data,a1
145:         move.w  #00,GR_Xpoint(a1)
146:         move.w  #00,GR_Ypoint(a1)
147:         move.w  #32,MV_Xsize(a1)
148:         move.w  #32,MV_Ysize(a1)
149:         move.w  #00,BG_Xpoint(a1)
150:         move.w  #00,BG_Ypoint(a1)
151:         jsr     GRAPHtoBG           * グラフィックをBGにセットする
152:
153:         SPON
154:
155: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
156: *
157: *                          割り込み設定
158: *
159:         DI                          * 割り込み禁止
160:
161:         moveq.l #1,d1               * V-BLANK割り込み設定
162:         lea.l   vblankint,a1
163:         IOCS    _VDISPST
164:         tst.l   d0
165:         bne     quit
166:
167:         EI                          * 割り込み許可
168:
169: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
170: *
171: *                          メイン処理ループ
172: *
173: mainloop:
174:         tst.w   b_blank             * V_BLANKの判定
175:         beq     mainloop            * メイン処理は1SYNC中1回しかやらない
176:         move.w  #0,b_blank          * 判定リセット
177:
178: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
179: *
180: *       USER PROCESS
181: *
182: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
183:
184:         move.w  #7,d1
185:         IOCS    _BITSNS             * ESCキーが押されるまでループ
186:
187:         move.l  #$4000,d1           * 加速度
188:
189: cur_L:  btst.l  #3,d0               * カーソル左
190:         bne     cur_U
191:         add.l   d1,X_MA
192: cur_U:  btst.l  #4,d0               * カーソル上
193:         bne     cur_R
194:         add.l   d1,Y_MA
195: cur_R:  btst.l  #5,d0               * カーソル右
196:         bne     cur_D
197:         sub.l   d1,X_MA
198: cur_D:  btst.l  #6,d0               * カーソル下
199:         bne     cur_end
200:         sub.l   d1,Y_MA
201: cur_end:
202:
203:         move.l  X_MA,d0
204:         add.l   d0,X_MS
205:         move.l  Y_MA,d0
206:         add.l   d0,Y_MS
207:
208:         move.w  X_MS,X_SCRL
209:         move.w  Y_MS,Y_SCRL
210:
211: gr_point:                           * グラフィックのスクロール座標計算
212:         move.w  X_SCRL,d0
213:         asr.w   #4,d0
214:         sub.w   #256-16,d0
215:         move.w  d0,GSR_dat+0
216:         move.w  d0,GSR_dat+4
217:         move.w  d0,GSR_dat+8
218:         move.w  d0,GSR_dat+12
219:
220:         move.w  Y_SCRL,d0
221:         asr.w   #4,d0
222:         sub.w   #256-16,d0
223:         move.w  d0,GSR_dat+2
224:         move.w  d0,GSR_dat+6
225:         move.w  d0,GSR_dat+10
226:         move.w  d0,GSR_dat+14
227:
228: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
229: *
230: *       処理の振り分け判定処理
231: *
232:         clr.l   d0                  * インデッスレジスタに使用
233: Xold_cmp:
234:         move.w  X_SCRL,d1
235:         sub.w   X_SCRLold,d1
236:         bmi     Xold_cmp00
237:         beq     Yold_cmp            * 変化がなかったらbitセットなし
238:         * プラスだったら右書き替え
239:         bset    #5,d0
240:         bra     Yold_cmp
241: Xold_cmp00:
242:         * マイナスだったら左書き替え
243:         bset    #4,d0
244: *       bra     Yold_cmp
245: Yold_cmp:
246:         move.w  d1,d3               * X座標変化量を退避
247:         move.w  Y_SCRL,d1
248:         sub.w   Y_SCRLold,d1
249:         bmi     Yold_cmp00
250:         beq     oldcmp_end          * 変化がなかったらbitセットなし
251:         * プラスだったら下書き替え
252:         bset    #3,d0
253:         bra     oldcmp_end
254: Yold_cmp00:
255:         * マイナスだったら上書き替え
256:         bset    #2,d0
257: *       bra     oldcmp_end
258: oldcmp_end:
259:         move.w  d1,d4               * Y座標変化量を退避
260:         tst.w   d0                  * 移動がなかったら処理無し
261:         beq     mainquit
262:
263:         tst.w   d3                  * X変化量の絶対値
264:         bpl     X_pl
265:         neg.w   d3
266: X_pl:
267:         tst.w   d4                  * Y変化量の絶対値
268:         bpl     Y_pl
269:         neg.w   d4
270: Y_pl:
271:         asr.w   #4,d3
272:         asr.w   #4,d4
273:         cmpi.w  #32,d3              * X座標が画面外だったら全面書き替え
274:         bls     X_noover
275:         clr.w   d0
276: X_noover:
277:         cmpi.w  #32,d4              * Y座標が画面外だったら全面書き替え
278:         bls     Y_noover
279:         clr.w   d0
280: Y_noover:
281:
282: jmp_sub:
283:         lea.l   jmp_addres,a0
284:         lea.l   GRAPHtoBG_data,a1
285:         move.w  X_SCRL,d1
286:         asr.w   #4,d1
287:         move.w  d1,GR_Xpoint(a1)
288:         move.w  Y_SCRL,d1
289:         asr.w   #4,d1
290:         move.w  d1,GR_Ypoint(a1)
291:         move.l  (a0,d0.w),a0
292:         move.w  #1,BGSET_busy      * BG書き替え中
293:         jsr     (a0)               * 専用のサブルーチンに飛ぶ
294:         move.w  #0,BGSET_busy      * BG書き替え終了
295:
296:         move.w  X_SCRL,X_SCRLold
297:         move.w  Y_SCRL,Y_SCRLold
298:
299: mainquit:
300:         move.w  #0,d1
301:         IOCS    _BITSNS             * ESCキーが押されるまでループ
302:         cmp.b   #%0000_0010,d0
303:         bne     mainloop
304:
```


▶9月号の安海さんへ，水戸には大型のパソコンショップはありませんが，駅前の石丸電気がいちばん品揃えがよいようです。場所は南町2丁目のバス停近く，ツルヤ書店の地下にあります。あとは駅前の川又書店の2FにTAKERUがあるのも要チェックです。今度は誰か私によいデートスポットを教えてください。 河合 章紀(23)茨城県

```
305: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
306: *
307: *                     終了処理
308: *
309: break:  DI
310:        suba.l  a1,a1
311:        IOCS    _VDISPST
312:
313:        move.l  #$10,d1         *  画面初期化
314:        IOCS    _CRTMOD
315:        IOCS    _G_CLR_ON
316:        IOCS    _OS_CURON       *  カーソルオン
317: quit:  EI                      *  割り込みを再許可
318:        DOS     _EXIT
319:
320: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
321: *
322: *                     割り込み処理
323: *
324: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
325: vblankint:
326:        move.w  #1,b_blank      *  判定セット
327:        tst.w   BGSET_busy      *  BGのデータセットの終了判定
328:        bne     vbi_quit        *  終わっていなかったらスクロール
329:                                *  レジスタを書き替えない
330:        SPOFF
331:        move.w  X_SCRL,BSR0_XPOS
332:        move.w  Y_SCRL,BSR0_YPOS
333:        SPON
334:
335:        move.w  GSR_dat+0,GSR0_XPOS
336:        move.w  GSR_dat+2,GSR0_YPOS
337:        move.w  GSR_dat+4,GSR1_XPOS
338:        move.w  GSR_dat+6,GSR1_YPOS
339:        move.w  GSR_dat+8,GSR2_XPOS
340:        move.w  GSR_dat+10,GSR2_YPOS
341:        move.w  GSR_dat+12,GSR3_XPOS
342:        move.w  GSR_dat+14,GSR3_YPOS
343:
344: vbi_quit:
345:        rte
346:
347: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
348: *
349: *               サブルーチン
350: *
351: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
352:
353: jmp_addres:                    *  RLDU
354:        .dc.l   ALL_set         *  0000
355:        .dc.l   UP_set          *  0001
356:        .dc.l   DW_set          *  0010
357:        .dc.l   0               *  0011
358:        .dc.l   LF_set          *  0100
359:        .dc.l   LU_set          *  0101
360:        .dc.l   LD_set          *  0110
361:        .dc.l   0               *  0111
362:        .dc.l   RG_set          *  1000
363:        .dc.l   RU_set          *  1001
364:        .dc.l   RD_set          *  1010
365:
366:
367: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*
368: *      全面書き替え    @@@
369: *                      @@@
370: *                      @@@
371: ALL_set:
372:        move.w  #32,MV_Xsize(a1)
373:        move.w  #32,MV_Ysize(a1)
374:        move.w  X_SCRL,d0
375:        asr.w   #4,d0
376:        move.w  d0,BG_Xpoint(a1)
377:        move.w  Y_SCRL,d0
378:        asr.w   #4,d0
379:        move.w  d0,BG_Ypoint(a1)
380:
381:        jsr     GRAPHtoBG
382:        rts
383: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*
384: *      上書き替え      @@@
385: *                      OOO
386: *                      OOO
387: UP_set:
388:        move.w  #32,MV_Xsize(a1)        *  X_SIZE指定
389:
390:        move.w  X_SCRL,d0
391:        asr.w   #4,d0
392:        move.w  d0,BG_Xpoint(a1)        *  X_書き替え位置指定
393:
394:        move.w  Y_SCRLold,d0
395:        sub.w   Y_SCRL,d0
396:        asr.w   #4,d0
397:        move.w  d0,MV_Ysize(a1)         *  Y_SIZE指定
398:
399:        move.w  Y_SCRL,d0
400:        asr.w   #4,d0
401:        move.w  d0,BG_Ypoint(a1)        *  Y_書き替え位置指定
402:
403:        jsr     GRAPHtoBG
404:        rts
405: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*
406: *      下書き替え      OOO
407: *                      OOO
408: *                      @@@
409: DW_set:
410:        move.w  #32,MV_Xsize(a1)        *  X_SIZE指定
411:
412:        move.w  X_SCRL,d0
413:        asr.w   #4,d0
414:        move.w  d0,BG_Xpoint(a1)        *  X_書き替え位置指定
415:
416:        move.w  Y_SCRL,d0
417:        sub.w   Y_SCRLold,d0
418:        asr.w   #4,d0
419:        move.w  d0,MV_Ysize(a1)         *  Y_SIZE指定
420:
421:        add.w   #32,GR_Ypoint(a1)       *  転送元Y座標計算
422:        sub.w   d0,GR_Ypoint(a1)
423:
424:        move.w  Y_SCRLold,d0
425:        asr.w   #4,d0
426:        add.w   #33,d0
427:        move.w  d0,BG_Ypoint(a1)        *  Y_書き替え位置指定
428:
429:        jsr     GRAPHtoBG
430:        rts
431: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*
432: *      左書き替え      @OO
433: *                      @OO
434: *                      @OO
435: LF_set:
436:        move.w  X_SCRLold,d0
437:        sub.w   X_SCRL,d0
438:        asr.w   #4,d0
439:        move.w  d0,MV_Xsize(a1)         *  X_SIZE指定
440:
441:        move.w  X_SCRL,d0
442:        asr.w   #4,d0
443:        move.w  d0,BG_Xpoint(a1)        *  X_書き替え位置指定
444:
445:        move.w  #32,MV_Ysize(a1)        *  Y_SIZE指定
446:
447:        move.w  Y_SCRL,d0
448:        asr.w   #4,d0
449:        move.w  d0,BG_Ypoint(a1)        *  Y_書き替え位置指定
450:
451:        jsr     GRAPHtoBG
452:        rts
453: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*
454: *      左上書き替え    @@@
455: *                      @OO
456: *                      @OO
457: LU_set:
458: LU_set01:
459:        move.w  #32,MV_Xsize(a1)        *  X_SIZE指定
460:
461:        move.w  X_SCRL,d0
462:        asr.w   #4,d0
463:        move.w  d0,BG_Xpoint(a1)        *  X_書き替え位置指定
464:
465:        move.w  Y_SCRLold,d0
466:        sub.w   Y_SCRL,d0
467:        asr.w   #4,d0
468:        move.w  d0,MV_Ysize(a1)         *  Y_SIZE指定
469:
470:        move.w  Y_SCRL,d0
471:        asr.w   #4,d0
472:        move.w  d0,BG_Ypoint(a1)        *  Y_書き替え位置指定
473:
474:        jsr     GRAPHtoBG
475:
476: LU_set02:
477:        move.w  X_SCRLold,d0
478:        sub.w   X_SCRL,d0
479:        asr.w   #4,d0
480:        move.w  d0,MV_Xsize(a1)         *  X_SIZE指定
481:
482:        move.w  X_SCRL,d0
483:        asr.w   #4,d0
484:        move.w  d0,BG_Xpoint(a1)        *  X_書き替え位置指定
485:
486:        move.w  Y_SCRLold,d0
487:        sub.w   Y_SCRL,d0
488:        add.w   #32*16,d0
489:        asr.w   #4,d0
490:        move.w  d0,MV_Ysize(a1)         *  Y_SIZE指定
491:
492:        move.w  Y_SCRL,d0
493:        asr.w   #4,d0
494:        move.w  d0,BG_Ypoint(a1)        *  Y_書き替え位置指定
495:
496:        jsr     GRAPHtoBG
497:        rts
498: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*
499: *      左下書き替え    @OO
500: *                      @OO
501: *                      @@@
502: LD_set:
503: LD_set01:
504:        move.w  #33,MV_Xsize(a1)        *  X_SIZE指定
505:
506:        move.w  X_SCRL,d0
507:        asr.w   #4,d0
508:        move.w  d0,BG_Xpoint(a1)        *  X_書き替え位置指定
509:
510:        move.w  Y_SCRL,d0
511:        sub.w   Y_SCRLold,d0
512:        asr.w   #4,d0
513:        move.w  d0,MV_Ysize(a1)         *  Y_SIZE指定
514:
515:        add.w   #32,GR_Ypoint(a1)       *  転送元Y座標計算
516:        sub.w   d0,GR_Ypoint(a1)
517:
518:        move.w  Y_SCRLold,d0
519:        asr.w   #4,d0
520:        add.w   #33,d0
521:        move.w  d0,BG_Ypoint(a1)        *  Y_書き替え位置指定
522:
523:        jsr     GRAPHtoBG
524:
525: LD_set02:
526:        move.w  X_SCRLold,d0
```

▶ムー，なんかOh!Xが薄くなったよーな。へっ，薄くなったのはワシの頭ですか？　う～，失礼しました（汗）。　藍原　和久(23)東京都

```
527:            sub.w    X_SCRL,d0
528:            asr.w    #4,d0
529:            move.w   d0,MV_Xsize(a1)            * X_SIZE指定
530:
531:            move.w   X_SCRL,d0
532:            asr.w    #4,d0
533:            move.w   d0,BG_Xpoint(a1)           * X_書き替え位置指定
534:
535:            move.w   Y_SCRLold,d0
536:            sub.w    Y_SCRL,d0
537:            asr.w    #4,d0
538:            add.w    #32,d0
539:            move.w   d0,MV_Ysize(a1)            * Y_SIZE指定
540:
541:            sub.w    d0,GR_Ypoint(a1)           * 転送元Y座標計算
542:
543:            move.w   Y_SCRL,d0
544:            asr.w    #4,d0
545:            move.w   d0,BG_Ypoint(a1)           * Y_書き替え位置指定
546:
547:            add.w    #1,BG_Ypoint(a1)           * 書き替え位置補正
548:
549:            jsr      GRAPHtoBG
550:            rts
551: *-*-*-*-*-*-*-*-*-*-*-*-*-*
552: *          右書き替え       OO@
553: *                            OO@
554: *                            OO@
555: RG_set:
556:            move.w   X_SCRL,d0
557:            sub.w    X_SCRLold,d0
558:            asr.w    #4,d0
559:            move.w   d0,MV_Xsize(a1)            * X_SIZE指定
560:
561:            add.w    #32,GR_Xpoint(a1)          * 転送元X座標計算
562:            sub.w    d0,GR_Xpoint(a1)
563:
564:            move.w   X_SCRLold,d0               * BG書き替え位置指定
565:            asr.w    #4,d0
566:            add.w    #33,d0
567:            move.w   d0,BG_Xpoint(a1)           * X_書き替え位置指定
568:
569:            move.w   #32,MV_Ysize(a1)           * Y_SIZE指定
570:
571:            move.w   Y_SCRL,d0
572:            asr.w    #4,d0
573:            move.w   d0,BG_Ypoint(a1)           * Y_書き替え位置指定
574:
575:            jsr      GRAPHtoBG
576:            rts
577: *-*-*-*-*-*-*-*-*-*-*-*-*-*
578: *          右上書き替え     @@@
579: *                            OO@
580: *                            OO@
581: RU_set:
582: RU_set01:
583:            move.w   #32,MV_Xsize(a1)           * X_SIZE指定
584:
585:            move.w   X_SCRL,d0
586:            asr.w    #4,d0
587:            move.w   d0,BG_Xpoint(a1)           * X_書き替え位置指定
588:
589:            move.w   Y_SCRLold,d0
590:            sub.w    Y_SCRL,d0
591:            asr.w    #4,d0
592:            move.w   d0,MV_Ysize(a1)            * Y_SIZE指定
593:
594:            move.w   Y_SCRL,d0
595:            asr.w    #4,d0
596:            move.w   d0,BG_Ypoint(a1)           * Y_書き替え位置指定
597:
598:            jsr      GRAPHtoBG
599:
600: RU_set02:
601:            add.w    #32,GR_Xpoint(a1)          * 転送元X座標計算
602:
603:            move.w   X_SCRL,d0
604:            sub.w    X_SCRLold,d0
605:            asr.w    #4,d0
606:            move.w   d0,MV_Xsize(a1)            * X_SIZE指定
607:
608:            move.w   X_SCRLold,d0
609:            asr.w    #4,d0
610:            add.w    #33,d0
611:            move.w   d0,BG_Xpoint(a1)           * X_書き替え位置指定
612:
613:            move.w   Y_SCRLold,d0
614:            sub.w    Y_SCRL,d0
615:            asr.w    #4,d0
616:            add.w    #32,d0
617:            move.w   d0,MV_Ysize(a1)            * Y_SIZE指定
618:
619:            move.w   Y_SCRL,d0
620:            asr.w    #4,d0
621:            move.w   d0,BG_Ypoint(a1)           * Y_書き替え位置指定
622:
623:            jsr      GRAPHtoBG
624:            rts
625: *-*-*-*-*-*-*-*-*-*-*-*-*-*
626: *          右下書き替え     OO@
627: *                            OO@
628: *                            @@@
629: RD_set:
630: RD_set01:
631:            move.w   #33,MV_Xsize(a1)           * X_SIZE指定
632:
633:            move.w   X_SCRL,d0
634:            asr.w    #4,d0
635:            move.w   d0,BG_Xpoint(a1)           * X_書き替え位置指定
636:
637:            move.w   Y_SCRL,d0
638:            sub.w    Y_SCRLold,d0
639:            asr.w    #4,d0
640:            move.w   d0,MV_Ysize(a1)            * Y_SIZE指定
641:
642:            add.w    #32,GR_Ypoint(a1)          * 転送元Y座標計算
643:            sub.w    d0,GR_Ypoint(a1)
644:
645:            move.w   Y_SCRLold,d0
646:            asr.w    #4,d0
647:            add.w    #33,d0
648:            move.w   d0,BG_Ypoint(a1)           * Y_書き替え位置指定
649:
650:            jsr      GRAPHtoBG
651:
652: RD_set02:
653:            add.w    #32,GR_Xpoint(a1)          * 転送元X座標計算
654:
655:            move.w   X_SCRL,d0
656:            sub.w    X_SCRLold,d0
657:            asr.w    #4,d0
658:            move.w   d0,MV_Xsize(a1)            * X_SIZE指定
659:
660:            move.w   X_SCRLold,d0
661:            asr.w    #4,d0
662:            add.w    #33,d0
663:            move.w   d0,BG_Xpoint(a1)           * X_書き替え位置指定
664:
665:            move.w   Y_SCRLold,d0
666:            sub.w    Y_SCRL,d0
667:            asr.w    #4,d0
668:            add.w    #32,d0
669:            move.w   d0,MV_Ysize(a1)            * Y_SIZE指定
670:
671:            sub.w    d0,GR_Ypoint(a1)           * 転送元Y座標計算
672:
673:            move.w   Y_SCRL,d0
674:            asr.w    #4,d0
675:            move.w   d0,BG_Ypoint(a1)           * Y_書き替え位置指定
676:
677:            add.w    #1,BG_Ypoint(a1)           * 書き替え位置補正
678:
679:            jsr      GRAPHtoBG
680:            rts
681:
682: *-*-*-*-*-*-*-*-*-*-*-*-**-*-*-*-*-*-*-*-*-*-*-*
683: * GRAPHtoBG
684: *          G-VTAMを指定の座標から指定のサイズだけ
685: *          BGのデータエリアに転送します
686: *
687: *          引 数
688: *          a1.l     パラメータのアドレス
689: *                 パラメータフォーマット
690: *                    +00.w    グラフィックGET用X座標
691: *                    +01.w    グラフィックGET用Y座標
692: *                    +02.w    転送するXサイズ-1
693: *                    +03.w    転送するYサイズ-1
694: *                    +04.w    BGPUT用X座標
695: *                    +05.w    BGPUT用Y座標
696: *
697: GRAPHtoBG:
698:            movem.l d0-d7/a0-a5,-(sp)
699:
700:            move.w   BG_Xpoint(a1),d0 * 書き替えX座標の補正
701:            sub.w    GR_Xpoint(a1),d0
702:            cmpi.w   #1,d0
703:            bne      XNH00
704:            sub.w    #1,BG_Xpoint(a1)
705:            add.w    #1,MV_Xsize(a1)
706: XNH00:
707:            tst.w    d0
708:            bpl      XNH01
709:            neg.w    d0
710:            add.w    d0,BG_Xpoint(a1)
711:            add.w    d0,MV_Xsize(a1)
712: XNH01:
713:
714:            move.w   BG_Ypoint(a1),d0 * 書き替えY座標の補正
715:            sub.w    GR_Ypoint(a1),d0
716:            cmpi.w   #1,d0
717:            bne      YNH00
718:            sub.w    #1,BG_Ypoint(a1)
719: YNH00:
720:            tst.w    d0
721:            bpl      YNH01
722:            neg.w    d0
723:            add.w    d0,BG_Ypoint(a1)
724: YNH01:
725:
726:            lea.l    GVRAM_ADDRES,a2
727:            lea.l    BG0_VRAM,a3
728:
729:            clr.l    d0                  * グラフィック Xカウンタ初期化
730:            move.l   d0,d1               * グラフィック Yカウンタ初期化
731:            move.l   d0,d2               * BG Xカウンタ初期化
732:            move.l   d0,d3               * BG Yカウンタ初期化
733:
734:            * GRAPH-VRAMのカウンタ計算
735:            move.w   GR_Xpoint(a1),d0
736:            add.l    d0,d0               * X座標を 2倍する
737:            andi.l   #(MAP_XSIZE-1)*2,d0
738:            move.w   GR_Ypoint(a1),d1
739:            asl.l    #8,d1               * Y座標を(1024*2)倍する
740:            asl.l    #3,d1
741:            andi.l   #(MAP_YSIZE-1)*2*MAP_YSIZE,d1
742:            * BG-VRAMのカウンタ計算
743:            move.w   BG_Xpoint(a1),d2
744:            add.l    d2,d2               * X座標を 2倍する
745:            andi.l   #(64-1)*2,d2
746:            move.w   BG_Ypoint(a1),d3
747:            asl.l    #7,d3               * Y座標を(64*2)倍する
748:            andi.l   #(64-1)*2*64,d3
```

▶DōGAからの電話を「変な団体」と間違えたのはうちの母のことです。かまたさん，ごめんなさい。ちゃんと説明しておきました。　　鈴木　健児(23)愛知県

```
749:
750:        move.l  d0,d4               * グラフィックのXカウンタを一時保持
751:        move.l  d2,d5               * BGのXカウンタを一時保持
752:
753:        move.w  MV_Ysize(a1),d6
754: move_loop00:
755:
756:        move.w  MV_Xsize(a1),d7
757: move_loop01:
758:        movea.l a2,a4
759:        movea.l a3,a5
760:
761:        add.l   d0,a4
762:        add.l   d1,a4
763:        add.l   d2,a5
764:        add.l   d3,a5
765:
766:        move.w  (a4),(a5)
767:        add.w   #$100,(a5)          * パレットを足す
768:
769:        addq.l  #2,d0
770:        andi.l  #(MAP_XSIZE-1)*2,d0
771:        addq.l  #2,d2
772:        andi.l  #(64-1)*2,d2
773:
774:        dbra    d7,move_loop01
775:
776:        move.l  d4,d0
777:        move.l  d5,d2
778:
779:        add.l   #MAP_XSIZE*2,d1
780:        andi.l  #(MAP_YSIZE-1)*2*MAP_YSIZE,d1
781:        add.l   #64*2,d3
782:        andi.l  #(64-1)*2*64,d3
783:
784:        dbra    d6,move_loop00
785:
786:        movem.l (sp)+,d0-d7/a0-a5
787:        rts
788:
789: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
790: *
791: *                   データエリア
792: *
793: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
794:        .data
795:        .even
796: b_blank          .ds.w   1         * V-BLANKの判定
797:
798: GRAPHtoBG_data: .ds.w   6         * GRAPHtoBGのパラメータワーク
799: GSR_dat:         .ds.w   8         * グラフィックスクロールレジスタワーク
800:
801:        * メインルーチンで使用
802: X_SCRL:          .ds.w   1         * 現在のX座標
803: Y_SCRL:          .ds.w   1         * 現在のY座標
804: X_SCRLold:       .ds.w   1         * 前回のX座標
805: Y_SCRLold:       .ds.w   1         * 前回のY座標
806: BGSET_busy:      .ds.w   1         * BGのデータセット終了判定
807:                                    * 0=セット終了  1=セット中
808:
809: X_MA:            .ds.l   1         * スクロール移動幅 16bit固定小数点
810: Y_MA:            .ds.l   1         * スクロール移動幅 16bit固定小数点
811: X_MS:            .ds.l   1         * スクロール値 16bit固定小数点
812: Y_MS:            .ds.l   1         * スクロール値 16bit固定小数点
813:
814: sp_scrdat:       * スプライトの転送データ
815:        .dc.w   256,256,%00_00_0001_00000001,%011
816:        .dc.w   256+16,256,%01_00_0001_00000001,%011
817:        .dc.w   256,256+16,%10_00_0001_00000001,%011
818:        .dc.w   256+16,256+16,%11_00_0001_00000001,%011
```

リスト2 GDISP_TEST.S

```
  1: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
  2: *
  3: *BGのデータエリア疑似拡張テスト用グラフィック生成プログラム
  4: *
  5: *1024*1024ドットの範囲に16色データを表示します
  6: *
  7: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
  8:
  9:        .include        iocscall.mac
 10:        .include        doscall.mac
 11:
 12: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
 13: *
 14: *                   定数&マクロ設定
 15: *
 16: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
 17:
 18: GRPH_PALET      equu    $e82000 * グラフィックパレット
 19:
 20: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
 21: *
 22: *                   メインプログラム
 23: *
 24: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
 25:        .text
 26:        .even
 27: init:
 28:        move.l  #$00,d1             * 512*512 1024*1024 16色
 29:        IOCS    _CRTMOD
 30:        IOCS    _G_CLR_ON
 31:        sub.l   a1,a1
 32:        IOCS    _B_SUPER            * スーパーバイザモードへ移行
 33:
 34:        move.w  #0000,d1
 35:        move.w  #0000,d2
 36:        move.w  #1023,d3
 37:        move.w  #1023,d4
 38:        IOCS    _WINDOW
 39:
 40: col_copy:      * グラフィックパレットをセットする
 41:        lea.l   col_data,a1
 42:        lea.l   GRPH_PALET,a2
 43:        move.w  #16-1,d0
 44: copy_loop:
 45:        move.w  (a1)+,(a2)+
 46:        dbra    d0,copy_loop
 47:
 48: symset:
 49:        move.w  #32-1,d3
 50: symset00:
 51:        move.w  d3,d2               * Y座標計算
 52:        asl.w   #5,d2
 53:        move.w  d2,symdat+2
 54:
 55:        move.w  #32-1,d4
 56: symset01:
 57:        move.w  d4,d2               * X座標計算
 58:        asl.w   #5,d2
 59:        add.w   #8,d2
 60:        move.w  d2,symdat+0
 61:
 62:        move.w  d4,d2               * カラーナンバー計算
 63:        andi.w  #4-1,d2
 64:        add.w   #1,d2
 65:        move.w  d2,symdat+10
 66:
 67:        lea.l   textdat,a1          * 表示文字計算
 68:        move.b  (a1,d4.w),symtext
 69:
 70:        lea.l   symdat,a1
 71:        IOCS    _SYMBOL
 72:
 73:        dbra    d4,symset01
 74:        dbra    d3,symset00
 75:
 76:        move.w  #32-1,d3
 77: symset02:
 78:        move.w  d3,d2               * Y座標計算
 79:        asl.w   #5,d2
 80:        move.w  d2,symdat+0
 81:
 82:        move.w  #32-1,d4
 83: symset03:
 84:        move.w  d4,d2               * X座標計算
 85:        asl.w   #5,d2
 86:        move.w  d2,symdat+2
 87:
 88:        move.w  d4,d2               * カラーナンバー計算
 89:        andi.w  #4-1,d2
 90:        add.w   #1+4,d2
 91:        move.w  d2,symdat+10
 92:
 93:        lea.l   textdat,a1          * 表示文字計算
 94:        move.b  (a1,d4.w),symtext
 95:
 96:        lea.l   symdat,a1
 97:        IOCS    _SYMBOL
 98:
 99:        dbra    d4,symset03
100:        dbra    d3,symset02
101:
102:        DOS     _EXIT
103:
104: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
105: *
106: *                   データエリア
107: *
108: *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
109:
110: *textdat:        .dc.b   "VUTSRQPONMLKJIHGFEDCBA9876543210"
111: textdat:         .dc.b   "0123456789ABCDEFGHIJKLMNOPQRSTUV"
112:
113: symtext:         .dc.b   00,00
114: symdat:          .dc.w   00,00
115:                  .dc.l   symtext
116:                  .dc.b   1,1
117:                  .dc.w   00
118:                  .dc.b   1,0
119:
120: col_data:        * BGのパレットに設定するカラー  (G_R_B_I)
121:                  .dc.w   %00000_00000_00000_0    * No.0
122:                  .dc.w   %10111_00000_00000_0    * No.1
123:                  .dc.w   %00000_10111_00000_0    * No.2
124:                  .dc.w   %00000_00000_10111_0    * No.3
125:                  .dc.w   %10111_10111_00000_0    * No.4
126:                  .dc.w   %11111_01111_01111_0    * No.5
127:                  .dc.w   %01111_11111_01111_0    * No.6
128:                  .dc.w   %01111_01111_11111_0    * No.7
129:                  .dc.w   %11111_11111_01111_0    * No.8
130:                  .dc.w   %00000_00000_00000_0    * No.9
131:                  .dc.w   %00000_00000_00000_0    * No.10
132:                  .dc.w   %00000_00000_00000_0    * No.11
133:                  .dc.w   %00000_00000_00000_0    * No.12
134:                  .dc.w   %00000_00000_00000_0    * No.13
135:                  .dc.w   %00000_00000_00000_0    * No.14
136:                  .dc.w   %00000_00000_00000_0    * No.15
137:
```

▶CD-ROM，MOの特集をやってください。　　　稲垣　治央(42)愛知県

BACK ISSUES

バックナンバー案内

ここには1993年10月号から1994年9月号までをご紹介しました。現在1993年9～12月号，1994年1，3～9月号の在庫がございます。バックナンバーはお近くの書店にご注文ください。定期購読の申し込み方法は152ページを参照してください。

1993

10月号
特別企画　秋祭りPRO-68K

連載　ハードコア3D/Computer Music入門/マシン語プログラミング
DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/吾輩はX68000である

- 特別付録　秋祭りPRO-68K（5"2HD）
- SCSIパックンTOWER JACK

LIVE in '93　未来予想図II/OutRunより　PASSING BREEZE
THE SOFTOUCH　コットン/The World of X68000/あにまーじゃんV3
全機種共通システム　シューティングゲームコアシステム作成法(4)

11月号
特集　ポリゴナイザSLASHの活用

連載　ハードコア3D/Computer Music入門/ファイル共有の実験と実践
こちらシステムX探偵事務所/目指せジョイスティックの星
響子 in CGわ～るど/ショートプロ/大人のためのX68000

- 新製品紹介　Easydraw SX-68K
  OS-9 Ultra C/Technical Tool Kit

LIVE in '93　渚のアデリーヌ/エロティカ・セブン
THE SOFTOUCH　ぶたさん/ダイアット・ヴァークス
全機種共通システム　S-OSで学ぶZ80マシン語講座(1)

12月号
特集　古今東西ゲーム議論

連載　ハードコア3D/マシン語プログラミング/響子 in CGわ～るど
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/Computer Music入門/ファイル共有の実験と実践

- 新製品紹介　MATIER ver.2.0
  C Compiler PRO-68K ver.2.1 NEW KIT

LIVE in '93　クリスマス・イブ/星に願いを
THE SOFTOUCH　ネメシス'90改/項劉記/スーパーリアル麻雀PII＆PIII
全機種共通システム　エディタアセンブラREDA再掲載

1994

1月号
特集　Z-MUSICシステムver.2.0

連載　ハードコア3D/ゲーム作りのKNOW HOW/響子 in CGわ～るど
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/Computer Music入門/ファイル共有の実験と実践

- 特別企画　ANOTHER CG WORLD in Hong Kong

LIVE in '94　LAST WAVE/スターウォーズ/明日への扉/夢路より　他
THE SOFTOUCH　ストリートファイターIIダッシュ/餓狼伝説2/
ドラゴンバスター/X68000傑作ゲーム選
全機種共通システム　S-OSで学ぶZ80マシン語講座(2)

2月号（品切れ）
特集　X-BASICとグラフィック

連載　ハードコア3D/ワンチップIC/響子 in CGわ～るど
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/Computer Music入門/ANOTHER CG WORLD

- 新製品紹介　ハイパーピクセルワークス

LIVE in '94　ランス3/新宿駅，巣鴨駅の発車メロディ/ピコー・ソング
THE SOFTOUCH　キーパー/マッドストーカーX68/餓狼伝説2　他
全機種共通システム　S-OSで学ぶZ80マシン語講座(3)
YGCSver.0.20リファレンスマニュアル

3月号
特別企画　ひなまつりPRO-68K

連載　ハードコア3D/マシン語プログラミング/ゲーム作りのKNOW HOW
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/響子 in CGわ～るど/ファイル共有の実験と実践

- 特別付録　ひなまつりPRO-68K（5"2HD）
- 新製品紹介　ビデオPC for X680x0

LIVE in '94　THEME FROM WINNING RUN/スターフォースアレンジ版
THE SOFTOUCH　卒業/マッドストーカーX68/B-FIELD!　他
全機種共通システム　S-OSで学ぶZ80マシン語講座(4)

4月号
特集　SX-WINDOWの活用

連載　ハードコア3D/こちらシステムX探偵事務所
DōGA CGアニメーション講座/響子 in CGわ～るど
ショートプロ/ローテク工作/ANOTHER CG WORLD

- 決定!　1993年度GAME OF THE YEAR
- 新製品紹介　ビデオ入力ユニットCZ-6VS1

LIVE in '94　宇宙戦艦ヤマト/プロジェクトA子
THE SOFTOUCH　ジオグラフシール/ぷよぷよ/レッスルエンジェルス2　他
全機種共通システム　S-OSで学ぶZ80マシン語講座(5)

5月号
特別企画　こいのぼりPRO-68K
第9回言わせてくれなくちゃだワ

連載　ハードコア3D/響子 in CGわ～るど/ショートプロ
DōGA CGアニメーション講座/ファイル共有の実験と実践
こちらシステムX探偵事務所/ANOTHER CG WORLD

- 特別付録　こいのぼりPRO-68K（5"2HD）
- 新製品紹介　WorkroomSX-68K/開発キットツール集

LIVE in '94　ロード/時間旅行
THE SOFTOUCH　大魔界村/アルゴスの戦士/ジオグラフシール　他

6月号
特集　X68000と仲間たち

連載　ハードコア3D/響子 in CGわ～るど/ショートプロ
ローテク工作/ファイル共有の実験と実践
こちらシステムX探偵事務所/ANOTHER CG WORLD

- 第5回Oh!Xアンケート分析大会
- 新製品紹介　F-Calc for x68k

LIVE in '94　キャミイのテーマ/The End of Love
THE SOFTOUCH　スーパーリアル麻雀PIV/あすか120% BURNING Fest.他
全機種共通システム　YGCS ver.0.30

7月号
特集　入門コンピュータミュージック

連載　響子 in CGわ～るど/ショートプロ/ゲーム作りのKNOW HOW
ローテク工作/システムX探偵事務所/マシン語プログラミング
DōGA CGアニメーション講座/ファイル共有の実験と実践

- 特別付録　CGA入門キット「GENIE」
- 実用講座　Photo CDでカードを作る

LIVE in '94　宇宙刑事ギャバン/究極戦隊ダダンダーン/スティング　他
THE SOFTOUCH　麻雀航海記/雀神クエスト/The World of X68000 II 他
全機種共通システム　シューティングゲーム作成講座(1)

8月号
特集　Graphic Movement

連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
ローテク工作/ANOTHER CG WORLD/善バビ
DōGA CGアニメーション講座/石の言葉，言葉の夢

- 新製品紹介　X-SIMM VI/Mu-1 GS
  SX-WINDOW ver.3.1

LIVE in '94　PURE GREEN/Ridge racer（POWER REMIX）
THE SOFTOUCH　Mr.Do!/Mr.Do! vs UNICORNS/レッスルエンジェルス3
全機種共通システム　シューティングゲーム作成講座(2)

9月号
特集　SX-WINDOW環境セットアップ

連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
ローテク工作/DōGA CGアニメーション講座/善バビ
システムX探偵事務所/ファイル共有の実験と実践

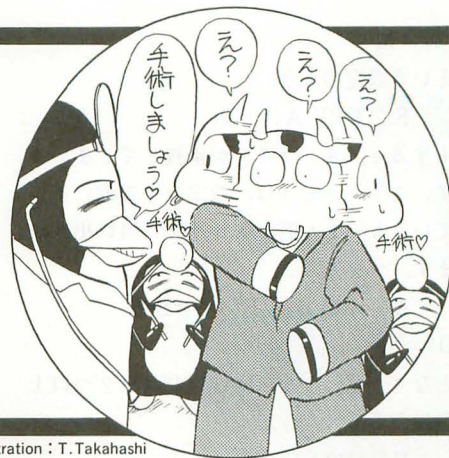
- 新製品紹介　X68030 D'ash/MJ-700V2C
- 新刊紹介　X680x0 TeX

LIVE in '94　LOVE IS ALL/HELL HOUND/踏切の通過音
THE SOFTOUCH　餓狼伝説SPECIAL
全機種共通システム　怪しいZ80の使い方（テクニック編）

(で)のショートプロぱーてぃ――その61

ああ,今年の誕生日……

Komura Satoshi 古村 聡

今月はドライブに関するツールが2本にゲームが1本。ゲームはけっこう手強いので心して挑戦してみてください。今年も(で)氏の誕生日の季節がやってきました。ところがなにか様子が変です。いったいなにが(で)氏に起こったのでしょう。

illustration：T. Takahashi

わはははは。発売日の翌日に今月号を買って読んでいるキミはラッキーである。なぜなら9月18日は私の誕生日だからなのである。皆で祝いたまえ,私もアンミラでミニミニなエプロンスカートねーちゃんたちに囲まれて祝っていることであろー,あ,ロイヤルのバニーさんもいいな,わはは……なんて当初の原稿では書いて「ちきしょー,1カ月先の俺ェ～,うらやましすぎるぞ」なんて思ってたんですけどねぇ。

先日,ノドの調子がちょっとおかしいんで病院へ行ったらいきなり「口にできものができてます。手術するので入院してください」ということになり,8月末に手術,誕生日はベッドの上で過ごすことになってしまいした。まったく俺ってヤツぁ……。誕生日と日曜日が重なることなんてそうそうないのに～。アンミラのおねーちゃんたちに会えないし,原稿も書き直すハメになるしさ～,ブチブチ。

ああ～,おいら,これでどうなるのかなぁ? これでなんかあったら「死ぬまでアンミラに行きたがった男」とか伝説になってしまうのだろうか? うう……イヤすぎる。い,いや,その前に押し入れの中のムフフな本や昔描きかけたままほってある漫画の原稿とか,日記とかぜ～んぶ処分しなくては! ひょえ～,絶対死ぬに死ねないぞ,こりゃ。

常連さんのおかえりなのだ

では今月1本目のプログラムを紹介しましょう。まずはデバイス名でドライブの割り当てをするプログラム,SETDRV.Xです。どうぞ。

SETDRV.X for X680x0

(要C Compiler ver.2.0以上)

大阪府 野崎哲也

X68000の場合,ハードディスクから起動するとハードディスクのユニット番号0番がA:,1番がB:と順番が決められてしまってますよね。でも,これだと,つながっているMOドライブの電源が切れていたりするとフロッピーのドライブ名がずれてしまったりとちょっと不便です。driveコマンドでドライブの順番を入れ換えてもいいんですが,driveコマンドでは1回で2つのドライブ名を入れ換えるだけですから,全部に好きなドライブ名を割り当てるのはちょっと大変です。そこで登場するのが,デバイス名で直接ドライブ名を指定できるこのツール,SETDRV.Xです。

このSETDRV.XはCのソースリストの形で掲載されています。まず,リスト1をED.Xなどのエディタで打ち込み,SETDRV.Cという名前でセーブしてください。それから,「C Compiler ver.2.0」以降のものでコンパイルします。

A>CC SETDRV.C

これで実行ファイルであるSETDRV.Xができました。

さて,それではこのプログラムの使い方。このプログラムではドライブの名前を,

(ドライブ名):=(デバイス記号)(ユニット番号)

という形式でオプションをつけることでドライブ名を変更します。たとえば,フロッピーをA:,B:にしたいときは,

A>SETDRV A:=F0 B:=F1

とすればOKです。デバイス記号は,

F　フロッピーディスク
R　RAMDISK
H　SASI
S　SCSI
M　MO
E　その他

となっています。デバイス記号が省略形ですが,なんとなくわかりますよね。引数はいくつでも指定できます。なにもオプションをつけずに,

SETDRV

とすると,現在のドライブの割り当て状況が表示されます。オプションスイッチ -nをつければ,画面に表示をさせないようにできます。AUTOEXEC.BATに入れておくときに便利ですね。

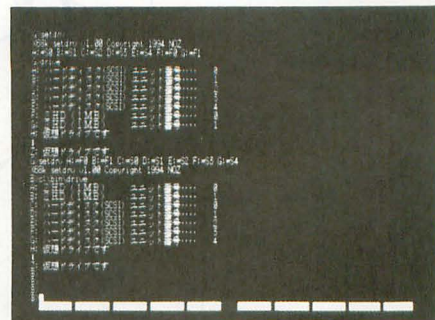
それから,割り当てたデバイスが存在しなかったとき,そのオプションは無視されます。たとえば,

SETDRV A:=F0 B:=F1 M:=M0 R:=R0

としたときにMOがなければ,その設定のところだけ無視して,

SETDRV A:=F0 B:=F1 R:=R0

を実行した場合と同じになるわけです。それから,

A:=F0 B:=F1 C:=S0

SETDRV.X

M：=*M0*
という設定のときに,

SETDRV A：=M0

とすると，Aドライブが重複してしまいます。こういうときは，設定をする前に空いているもっとも若いドライブに追い出されます。つまり，

A：=M0 B：=F1 C：=S0

D：=F0

となります。また，RAMDISKを２つ以上登録すると，

R：=R0 S：=R0

となりますが，このときに，

SETDRV A：=R0

としたときは，若いRドライブが移動の対象になり，Sドライブを移動させる手段はありません。ま，仕様だってことで……。

あらまぁ，野崎さんではありませんか。お元気でしたか？　なになに？「それにしても，昔は常連とまでいってくださいましたが，覚えているでしょうか？（以下略）」だそうで，もちろん覚えてますとも。ここしばらく，野崎さんのプログラムを見ないあいだにX68000をとりまく環境もずいぶん変わりましたね。CD-ROMドライバが発売されたり，MOドライブが一般的になったり，1Gバイトのハードディスクもそんなにめずらしくなくなりましたしね。実をいうと，私もMOやCD-ROMを使ったりするのでフロッピーをA，Bドライブに固定するのにこのプログラムを使ってます。いや，便利ですよ～。

このプログラムは通信で配布されている野崎さん作のフリーソフトウェアとまったく同じのものだそうです。それから，仮想ドライブ関連のDOSコールを使っているので「XC ver.1.0」のライブラリではコンパイルできませんので注意してください。

新人君のお手伝い

続いては１本目のSETDRV.Xと一緒に使うと役立ちそうなプログラムを選んでみました。このコーナー初投稿の渡辺さんの作品でVLSEARCH.Xです。どうぞ。

VLSEARCH.X for X680x0

（要C Compiler ver.2.0以上）

神奈川県　渡辺節弥

このプログラムは，X680x0につながっているドライブから，指定されたボリュームラベルを持つドライブを検索して，EXITコードで有無を返すプログラムです。オプションで環境変数を指定するとA：やB：といったドライブ名を設定できます。

このプログラムも１本目と同じようにCのソースリストの形で掲載されています。エディタで打ち込んで，VLSEARCH.Cというファイルを作り「C Compiler ver.2.0」以上で，

A>CC VLSEARCH.C

として実行ファイルVLSEARCH.Xを作ってください。

でもってお待たせの使い方ですね。このプログラムでは，

A>VLSEARCH ボリュームラベル名

とプログラム名のあとにボリュームラベル名を書いて，その名前のドライブを探させます。

A>VLSEARCH RAM_DISK

とすると，「ボリュームラベルRAM_DISKはドライブD：です」などと表示します。－eオプションをつけて環境変数を指定すると，その環境変数にボリュームラベルのあったドライブ名を格納します。

A>VLSEARCH.X　ASK68kDIC－e DIC

として，ASK68kDICというボリュームラベル名のドライブD：があれば，環境変数DICにD：が設定されます。

SETDRVと一緒に使うと便利そうでしょ。残念ながら，検索の途中で各ドライブにディスクが入っていないと「ディスクが入っていません，入れてください」と表示されたり，まだまだ甘さの目立つプログラムですが，そこはそれ，使う人が改造してしまいましょう。それでこそ，ソースの載ってるショートプロならではの使い方ってもんです。私も「ボリュームラベル名のあるドライブがあったらEXITコードで返す」プログラムにしようかな～，などと思っています。でもって，MOドライブ上に辞書が入っていたら，それを使って，ないときにはハードディスクの辞書を使うようなAUTOEXEC.BATを作って辞書を持ち歩こうかな～と思ったんだよね。なかなかぐ～なアイデアだと思いません？

シメはやっぱりゲームなのだ

さてさて，お待たせさま。最後のシメはやっぱりゲームですよね。この人ももう常連さん。平井さんの作品でDIKE.BASです。どうぞ。

DIKE.BAS for X680x0

（X-BASIC,コンパイル推奨,
ジョイスティック対応）

三重県　平井栄治

X-BASICのプログラムですので，例によってBASICを起動してから打ち込んでRUNで実行です。「私のマシン，遅いのよね～」という方は，

A>CC DIKE.BAS

でコンパイルしましょう。

緑色の自機をジョイスティックかキーボードで操ります。で，今回は迷路の中のどこかにいる赤い敵を捕まえてください。

スタートはリターンやスペースなどの適当なキーか，ジョイスティックのボタンを押してください。移動については，↑を押すと前進，↓を押すと後退，←を押すと左60度回転，→を押すと右60度回転します(ジョイスティックはそれぞれの方向に対応)。そして，緑六角形を赤六角形に重ねればゲームクリアです。Scoreは緑六角形が停止していると－1，回転すると－5，移動すると－10されていきます。Scoreが0になるとゲームオーバーです。また，画面に迷路の全体図が表示されていないときにESCを押すと，プログラムの実行を終了。HOMEを押すと，緑六角形を初期位置に復帰，CLRキーを押すと，High Scoreを0にし，新たな迷路を作成します。左右の回転については，キーボードではリピート可能にしてありますが，ジョイスティックではリピート不可にしてあります。

これから自動車の免許を取る人や，免許はあるけど運転に自信がない人は，DIKEをしない，あるいは，前進のみで後退はしないほうがいいと思います。車はバックしているときもハンドルを回したほうに曲がりますが，DIKEで←（→）を押した場合は，いつでも，左（右）60度回転します。つまり，車とDIKEでは後退(バック)時の感覚が異なるのです。車の感覚ができていない人がDIKEの感覚に慣れて実際の車を運転すると，バックでは普通スピードを出さないのでたいした事故は起きないでしょうが，車両や建造物をBREAKしちゃうので注意しましょう。

いやいやいや。ゲームの内容としては迷路のどこかにいる敵を捕まえるだけ，という簡単なものなんですけど，ジョイスティック，キーボード両方に対応してたり，画面が回転っぽくなっていたりして，常連さんの芸の細かさってのが出てます。もっとも，芸を追求するあまりリストが長くなりがちなのも常連さんの傾向なんだけど。

作者の平井さんの投稿原稿には，「前作のDIODEは慣れるまでは難しいのですが，DIKEは慣れても難しい。迷路自体は簡単なはずなのに，自分が迷路の中に入るとどこにいるかわからなくなっちゃうんです。迷ったら，右手法で動くようにしてますが

DIKE.BAS

……あうあう。誰か迷路を探索するいい思考方法があったら教えてくださいまし～！」と，あります。おいらも教えてほしいけど……むーん，病室にFAX持ち込んで，編集室からハガキを転送してもらおうかな。

さて，そんなわけで来月，私はお休みします。1,2回の休みですむのか，ずっと休むことになるのかわからないけど（だって，まだ良性か悪性かわかんないんだもん)，無事退院できたらまたお会いしましょ。誰か代打に立ってくれるのかな？

それではその日まで……その日がきますように。あでゅー☆

リスト1 SETDRV.C

```
  1: /***********************************************************/
  2:
  3:  デバイス名でドライブの割りあてをするプログラム ver 1.00
  4:
  5:                                 開発開始 平成6年4月12日
  6:                                 最終変更 平成6年5月 3 日
  7:
  8:                                        programmed by NOZ
  9:
 10: ***********************************************************/
 11: #include <stdio.h>
 12: #include <stdlib.h>
 13: #include <doslib.h>
 14:
 15: #define TITLE   "X68k setdrv v1.00 Copyright 1994 NOZ¥n"
 16:
 17: #define MAX_DRIVE ( 'Z' - 'A' + 1 )
 18:
 19: typedef enum {
 20:     FD=0xfe , RAM=0xf9 , SASI=0xf8 , SCSI=0xf7 ,
 21:     MO=0xf6 , ETC=0xf5 , OTHER
 22: } DRIVE_ID;
 23:
 24: typedef enum {
 25:     C_FD='F' , C_RAM='R' , C_SASI='H', C_SCSI='S' ,
 26:     C_MO='M' , C_ETC='E' , C_OTHER ='O'
 27: } DRIVE_TYPE;
 28:
 29: typedef struct DRIVE_TABLE {
 30:     DRIVE_ID        id;
 31:     DRIVE_TYPE      type;
 32: } DRIVE_TABLE;
 33:
 34: static DRIVE_TABLE      driveTable[] = {
 35:     { FD    , C_FD          } ,
 36:     { RAM   , C_RAM         } ,
 37:     { SASI  , C_SASI        } ,
 38:     { SCSI  , C_SCSI        } ,
 39:     { MO    , C_MO          } ,
 40:     { ETC   , C_ETC         } ,
 41:     { OTHER , C_OTHER       }
 42: };
 43:
 44:
 45: static DRIVE_TYPE convertIDtype( id )
 46: DRIVE_ID id;
 47: {
 48:     int i;
 49:
 50:     for ( i = 0 ; driveTable[i].id != OTHER ; i++ ) {
 51:         if (id == driveTable[i].id)
 52:             break;
 53:     }
 54:     return( driveTable[i].type );
 55: }
 56:
 57:
 58: static DRIVE_ID convertID( type )
 59: DRIVE_TYPE type;
 60: {
 61:     int i;
 62:
 63:     for ( i = 0 ; driveTable[i].id != OTHER ; i++ ) {
 64:         if (type == driveTable[i].type)
 65:             break;
 66:     }
 67:     return( driveTable[i].id );
 68: }
 69:
 70:
 71: static int checkDrive( drive )
 72: int drive;
 73: {
 74:     static char drv[] = "A:";
 75:     static char buf[256];
 76:     static struct DPBPTR dpb;
 77:
 78:     if (GETDPB( drive + 1 , &dpb ) != 0) {
 79:         *drv = drive + 'A';
 80:         if (GETASSIGN( drv , buf ) != 80)
 81:             return( 0 );
 82:     }
 83:     return( 1 );
 84: }
 85:
 86:
 87: static int searchDrive( id , unit )
 88: DRIVE_ID id;
 89: int unit;
 90: {
 91:     struct DPBPTR dpb;
 92:     int i;
 93:
 94:     for ( i = 0 ; i < MAX_DRIVE ; i++ ) {
 95:         if (id == 0) {
 96:             if (checkDrive( i ) == 0)
 97:                 return( i );
 98:             continue;
 99:         }
100:         if (GETDPB( i + 1 , &dpb ) != 0)
101:             continue;
102:         if ( (dpb.id == id) && (dpb.unit == unit) )
103:             return( i );
104:     }
105:     return( -1 );
106: }
```

```
107:
108:
109: static void drvXchg( a , b )
110: int a;
111: int b;
112: {
113:     DRVXCHG( a + 1 , b + 1 );
114:     if ( a == CURDRV() )
115:         CHGDRV( b );
116:     if ( b == CURDRV() )
117:         CHGDRV( a );
118: }
119:
120:
121: static int changeDrive( drive , id , unit )
122: int drive;
123: DRIVE_ID id;
124: int unit;
125: {
126:     int sourceDrive;
127:
128:     if ( (sourceDrive = searchDrive( id , unit )) == -1 )
129:         return( 1 );
130:
131:     if (sourceDrive == drive)
132:         return( 0 );
133:     if (checkDrive( drive ) != 0) {
134:         int tempDrive;
135:         if ( (tempDrive = searchDrive( 0 , 0 )) == -1 )
136:             return( 2 );
137:         drvXchg( tempDrive , drive );
138:     }
139:     drvXchg( sourceDrive , drive );
140:     return( 0 );
141: }
142:
143:
144: static void usage( void )
145: {
146:     fprintf( stderr , TITLE );
147:     puts( "使い方: setdrv [-n] <設定> ....¥n"
148:           "¥t-n¥t画面出力を禁止します¥n"
149:           "¥t¥t引数なしで起動すると、現在の設定状況を表示します" );
150: }
151:
152:
153: static void printDrive( void )
154: {
155:     int drv;
156:     struct DPBPTR dpb;
157:
158:     fprintf( stderr , TITLE );
159:     for ( drv = 1 ; drv <= MAX_DRIVE ; drv++ ) {
160:         if (GETDPB( drv  , &dpb ) != 0)
161:             continue;
162:         printf( "%c:=%c%d " , drv + 'A' - 1 , convertIDtype
( dpb.id ) , dpb.unit );
163:     }
164:     putchar( '¥n' );
165: }
166:
167:
168: extern void main( argc , argv )
169: int argc;
170: char *argv[];
171: {
172:     char drive;
173:     char type;
174:     int unit;
175:     int i = 1;
176:     int message = 1;
177:
178:     if (argc == 1) {
179:         printDrive();
180:         return;
181:     }
182:     if ( (*argv[1] == '-') || (*argv[1] == '/') ) {
183:         i++;
184:         if ( (argv[1][1] == 'n') || (argv[1][1] == 'N') )
185:             message = 0;
186:         else {
187:             usage();
188:             return;
189:         }
190:     }
191:
192:     if (message)
193:         fprintf( stderr , TITLE );
194:
195:     for ( ; i < argc ; i++ ) {
196:         if (sscanf( argv[i] , "%c:=%c%d" , &drive , &type ,
 &unit ) != 3) {
197:             fprintf( stderr , "'%s' is a syntax error.¥n" ,
 argv[i] );
198:             continue;
199:         }
200:         if ( ('A' <= drive) && (drive <= 'Z') )
201:             drive -= 'A';
202:         else {
203:             if ( ('a' <= drive) && (drive <= 'z') )
204:                 drive -= 'a';
205:             else {
206:                 fprintf( stderr , "'%s' is a syntax error.¥
n" , argv[i] );
207:                 continue;
208:             }
209:         }
210:         if (changeDrive(drive,convertID(type),unit) != 0) {
211:             if (message)
212:                 fprintf( stderr , "'%c%d' is not found.¥n"
, type , unit );
213:             continue;
214:         }
215:     }
216: }
```

リスト2 VLSEARCH.C

```
  1:
  2: /**********************************************************/
  3: /*                                                        */
  4: /*   指定されたボリュームラベルをもつドライブを検索する。     */
  5: /*                                                        */
  6: /**********************************************************/
  7:
  8: #include <stdio.h>
  9: #include <stdlib.h>
 10: #define  __DOS_INLINE__
 11: #include <sys¥dos.h>
 12:
 13: typedef  struct _filbuf  FILBUF;
 14:
 15: typedef  struct {
 16:     char  label[24];                  /* ボリュームラベル */
 17: } VOLTBL;
 18:
 19: void st_Param( int, char **, char *, char *, int * );
 20: void st_MakeTable( VOLTBL * );
 21:
 22: /*------------------------------------------------------*/
 23: /*                                                      */
 24: /*------------------------------------------------------*/
 25: void
 26: main( argc, argv )
 27: int   argc;
 28: char *argv[];
 29: {
 30:     int  i;                  /* ループ制御変数              */
 31:     int  rinf;               /* 関数値                      */
 32:     int  drive_no; /*ドライブ番号(A=0,B=1,…,Z=25,無=-1)    */
 33:     int  cmplen;             /* 文字比較数                  */
 34:     char  drive_string[8]; /* ドライブ名編集領域           */
 35:     char  vol_name[24];     /* ボリュームラベル             */
 36:     char  env_name[32];     /* 環境変数名                   */
 37:     VOLTBL  voltbl[26];     /* ボリュームテーブル           */
 38:
 39:     /* パラメータ解析 */
 40:     st_Param( argc, argv, vol_name, env_name, &cmplen );
 41:
 42:     /* テーブル作成 */
 43:     st_MakeTable( voltbl );
 44:
 45:     /* ボリュームラベルを検索する */
 46:     for( i = 0, drive_no = -1; i < 26; i++ ) {
 47:         if( strncmp(vol_name,voltbl[i].label,cmplen)==0 )
 48:             drive_no = i;
 49:             break;
 50:         }
 51:     }
 52:
 53:     /* ボリュームラベルが見つからなかった場合は */
 54:     /* EXITコード=1で終了する                   */
 55:     if( drive_no == -1 ) {
 56:         printf( "ボリュームラベル %s は", vol_name );
 57:         printf( "見つかりません¥n" );
 58:         exit( 1 );
 59:     }
 60:
 61:     sprintf( drive_string, "%c:", 'A' + drive_no );
 62:     printf( "ボリュームラベル %s は", vol_name );
 63:     printf( "ドライブ %s です¥n", drive_string )·
 64:
 65:     /* 環境変数が指定されているか? */
 66:     if( strlen(env_name) > 0 ) {
 67:
 68:         /* 見つかったラベルのドライブ名を */
 69:         /* (親の環境の)環境変数に設定する */
 70:         rinf = _dos_setenv( env_name, 0, drive_string );
 71:         if( rinf < 0 ) {
 72:             printf( "vlsearch: _dos_setenv error. rinf=%d¥n
", rinf );
 73:             exit( 2 );
 74:         }
 75:
 76:     }
 77:
 78:     exit( 0 );
 79: }
 80: /*------------------------------------------------------*/
 81: /*     パラメータ解析                                    */
 82: /*------------------------------------------------------*/
 83: void
 84: st_Param( argc, argv, vol_name, env_name, cmplen )
 85: int   argc;               /* アーギュメント数    (in/   )   */
 86: char *argv[];             /* アーギュメント      (in/   )   */
 87: char *vol_name;           /* ボリュームラベル    (   /out)  */
 88: char *env_name;           /* 環境変数名             (   /out)   */
 89: int  *cmplen;             /* 文字比較数             (   /out)   */
 90: {
 91:     int      i;           /* ループ制御変数                 */
 92:     int      rinf;        /* 関数値                         */
 93:     int      opt;         /* オプション文字                 */
 94:     int      errflag;     /* エラーフラグ                   */
 95:
 96:     /* 初期化 */
 97:     vol_name[0] = '¥0';
 98:     env_name[0] = '¥0';
 99:     *cmplen = 23;
100:
101:     /* getopt()のエラー出力を抑制する */
102:     opterr = 0;
103:
104:     /* (エラーフラグが立ったらループ終了) */
105:     for( errflag = 0; errflag == 0; ) {
```

▶先日，初めてコミケに行ってきました。「intel outside」と書いてあるTシャツを着て同人ソフトのブースへ行ったら，AMIGAのソフトを売っている人に笑われただけで，PC-98サークルの人に石を投げられなかったのでよかったです。　　磯崎　玄(18)千葉県

```
106:
107:         /* アーギュメントからオプション文字を取り出す */
108:         /* (オプション文字がなくなったらループ終了)   */
109:         if( ( opt=getopt( argc, argv, "e:E:8" ) )==EOF ) {
110:             break;
111:         }
112:
113:         /* オプション文字を解析する */
114:         switch( opt ) {
115:           case 'e':              /* 環境変数名指定                */
116:           case 'E':              /*                              */
117:             if( strlen(env_name) == 0 ) {
118:                 strcpy( env_name, optarg );
119:             } else {
120:                 errflag = 1;
121:             }
122:             break;
123:           case '8':              /* 8文字比較オプション            */
124:             *cmplen = 8;
125:             break;
126:           case '?':              /* オプション異常                */
127:           default:               /* その他                        */
128:             errflag = 1;
129:             break;
130:         }
131:
132:     }
133:
134:     /* アーギュメントが残っているか? */
135:     if( optind < argc ) {
136:         /* ボリュームラベルを格納 */
137:         strcpy( vol_name, argv[optind++] );
138:     /* 残っていない... */
139:     } else {
140:         errflag = 1;
141:     }
142:
143:     /* エラーフラグが立っているか、アーギュメントが */
144:     /* 残っていたら、使用方法を表示して終了する      */
145:     if( ( errflag == 1 ) || ( optind < argc ) ) {
146:         printf( "Usage: vlsearch volume [-e environ] [-8]¥n
" );
147:         printf( " パラメータ¥n" );
148:         printf( "        volume : 検索するボリュームラベル¥n" );
149:         printf( "    -e environ : 検索結果を格納する環境変数名¥n" );
150:         printf( "            -8 : チェック文字数を8文字とする¥n" );
151:         printf( " 終了コード(exitcode)¥n" );
152:         printf( "     0 : ボリュームラベル有¥n" );
153:         printf( "     1 : ボリュームラベル無¥n" );
154:         printf( "     2 : 内部処理エラー¥n" );
155:         printf( " 注意¥n" );
156:         printf( "     ・ボリュームラベルが無い時は、環境変数の¥n" );
157:         printf( "       定義(変更)は行いません。¥n" );
158:         printf( "     ・同一ラベルが複数ある場合、環境変数には¥n" );
159:         printf( "       一番若いドライブ名が格納されます。¥n" );
160:         exit( 2 );
161:     }
162:
163:     return;
164: }
165: /*-----------------------------------------------------*/
166: /*     テーブル作成                                      */
167: /*-----------------------------------------------------*/
168: void
169: st_MakeTable( voltbl )
170: VOLTBL   voltbl[26];           /* ボリュームテーブル (  /out)    */
171: {
172:     int      i;                /* ループ制御変数                  */
173:     int      rinf;             /* 関数値                          */
174:     char     file_name[16];    /* 検索ファイル名                  */
175:     FILBUF   filbuf;           /* ファイル情報                    */
176:
177:     /* 初期化 */
178:     memset( voltbl, 0, sizeof(voltbl) );
179:
180:     /* 各ドライブのボリュームラベルを集める */
181:     for( i = 0; i < 26; i++ ) {
182:
183:         /* ドライブが使用可能かを調べる */
184:         rinf = _dos_drvctrl( 0, i+1 );
185:         if( ( rinf & 0x07 ) != 0x02 ) {
186:             continue;
187:         }
188:
189:         /* "ボリュームID"のファイル属性を持つ */
190:         /* ファイルを検索する                  */
191:         sprintf( file_name, "%c:¥¥*.*", 'A'+i );
192:         rinf = _dos_files( &filbuf,file_name,_DOS_IFVOL );
193:         if( rinf < 0 ) {
194:             switch( rinf ) {
195:               case _DOSE_NOENT:                 /* ファイル無し   */
196:               case _DOSE_ILGDRV:                /* ドライブ異常   */
197:                 continue;
198:               default:
199:                 printf( "vlsearch: _dos_files error. rinf=%
d¥n", rinf );
200:                 exit( 2 );
201:             }
202:         }
203:
204:         /* ファイル名をテーブルに設定する */
205:         strcpy( voltbl[i].label, filbuf.name );
206:
207:     }
208:
209:     return;
210: }
```

リスト3 DIKE.BAS

```
10    screen 0,2,1,1:   palet(1,   48  ):window(0,0,395,256)
20          sp_init():   palet(3,16912  ):          m         =1
30           m_init():sp_color(1,   48,1):          n         =0
40       m_tempo(181):sp_color(1, 1536,2):          w         =1
50      m_alloc(1,81):sp_color(1,16912,3):dim char x(33152)
60     m_assign(1,1):sp_color(1,49152,4):dim       y(  168)
70   m_trk(1,"r16."):sp_disp(1):sp_on(0):dim       z(   35)={
80           146,218,128,236,110,218, 92,200, 74,182, 74,146,
90            74,110, 74, 74, 92, 56,110, 38,128, 20,146, 38,
100          164, 56,182, 74,182,110,182,146,182,182,164,200}
110   line(1,8,4, 1,1):line(4, 1,11, 1,1):line(14,8,11, 1,1)
120   line(1,7,4,14,1):line(4,14,11,14,1):line(14,7,11,14,1)
130    a():paint(7,7,1):a():wipe():bg_fill(0,0):bg_fill(1,0)
140  while n<4:b():f=1:g=256:o=0:p=0:u=74:v=182:apage(0)
150  vpage(1):locate 3,0:print"High score DIKE Score":c()
160        sp_set(0,0,0,0,3):bg_set(0,0,0):bg_set(1,1,0)
170  for h=0 to g:line(65-f,h,191+f,h,3,43690):f=1-f:next
180  for w=0 to 5:            for f=0 to 5:d(): next:next
190  for f=0 to 5:       if f=5 then{y(1)=0}else{y(1)=f+1}
200       y(0)=f:        if f=0 then{y(2)=5}else{y(2)=f-1}
210  for g=0 to 2: h=6*f+g+g:u=z(h):v=z(h+1):w=y(      0):d()
220  for i=0 to 3:                             w=y(rnd()*3):d()
230               next:next:next:n=1:get(64,0,192,256,x):f=0
240  for g=0 to 33152:if x(g)=1 then{f=f+1
250                  }else{}:next:h=f*rnd():i=(f-1)*rnd():f=0
260  for g=0 to 33152:if x(g)=1 then{f=f+1:if f=h then{x(g)=0
270      break}else{}}else{}:next:                         f=0
280  for h=0 to 33152:if x(h)=1 then{f=f+1:if f=i then{x(g)=2
290      break}else{}}else{}:next:       r=g:               s=h
300      t=6*rnd():wipe():while n<3:b():w=t:v=s/129:u=s-v*129
310      if n=2 and o<p then{o=p}else{}:if n=1 then{e():f=136
320 bg_set(0,0,1):bg_set(1,1,1):sp_set(0,f,144,1025,3)}else{}
330   c():n=0:vpage(0):locate 14,0:print"Push":beep:while n<1
340    if strig(1)<>0 then{break}else{}:switch asc(inkey$(0))
350 case 27:n=4:break:case 11:n=1:break:case 12:n=3:    break
360 case 28:            case 30:             case  0
370 case 29:            case 31:n=0:break:default:n=2:endswitch
380       if n=2 then{n=0:break}else{}:endwhile:while n<1:e()
390  f=0:q=0:if 1=1=1 then{if b_argc=1 then{}else{f=1}}else{}
400 if f=0 then{p=99999:locate 14,0:print"DIKE"
410       }else{p= 9999:locate 14,0:print"    ":vpage(2)}:c()
420       break:endwhile:while n<1:m_play(1):f=stick(1):p=p-1
430    if f=0 then{f=5}else{}:g=1-(f-1)/3:f=f+g*3-5:h=6
440        switch asc(inkey$(0)):case 28:    f=f+1:q=0:break
450 case 11:             n=1:break:case 29:    f=f-1:q=0:break
460 case 12:             n=3:break:case 30:    g=g-1:    break
470 case 27:             n=4:break:case 31:    g=g+1
480    endswitch:while n<1:b():if g=0 then{}else{if g<0 then{
490      if y(83)=3 then{}else{h=w  :p=p-9}}else{
500      if y(85)=3 then{}else{h=w-3:p=p-9:       if h<0 then{
510    h=h+6}else{}}}}: switch h:case 3:v=v+2:u=u  :h=5:break
520 case 0:v=v-2:u=u  :h=5:break:case 4:v=v+1:u=u-1:h=5:break
530 case 1:v=v-1:u=u+1:h=5:break:case 5:v=v-1:u=u-1:h=5:break
540 case 2:v=v+1:u=u+1:h=5:break:case 6:            h=4
550    endswitch:if f=0 then{q=0}else{
560  if f<0 then{if q=2 then{   }else{
570             if w=0 then{w=5}else{w=w-1}:p=p-h:q=2}}else{
580             if q=1 then{   }else{
590             if w=5 then{w=0}else{w=w+1}:p=p-h:q=1}}
600   }:if p<1 then{p=0:n=2}else{}
610 e():if y(84)=2 then{n=2}else{}:while m_stat()<>0:endwhile
620 c():   break:endwhile:endwhile:endwhile:endwhile:end
630 func a():get(0,0,15,15,x):sp_def(w,x,1):   w=w+1:endfunc
640 func b():              while inkey$(0)<>"":endwhile:endfunc
650 func c():                bg_scroll(0,128-128*m,504)
660   home(1,128-128*m,0):bg_scroll(1,132-128*m,504)
670      locate  8,1:print right$("      "+itoa(o),5)
680      locate 19,1:print  left$(itoa(p)+"      ",5):endfunc
690 func d():j=    1: switch w:case 0:   j=-j:case 3:    j=-j
700                    line(u-1  ,v+1*j,u- 1  ,v-19*j,1,43690)
710                    line(u    ,v+2*j,u     ,v-20*j,1,43690)
720                    line(u+1  ,v+1*j,u+ 1  ,v-19*j,1,43690)
730 v=v-18*j:u=u    :     break:case 1:   j=-j:case 4:    j=-j
740                    line(u-1*j,v-1*j,u+ 9*j,v-11*j,1      )
750                    line(u-1*j,v+1*j,u+10*j,v-10*j,1      )
760                    line(u    ,v+2*j,u+10*j,v- 8*j,1      )
770 v=v- 9*j:u=u+9*j:     break:case 2:   j=-j:case 5:    j=-j
780                    line(u    ,v-2*j,u+10*j,v+ 8*j,1      )
790                    line(u-1*j,v-1*j,u+10*j,v+10*j,1      )
800                    line(u-1*j,v+1*j,u+ 9*j,v+11*j,1      )
810 v=v+ 9*j:u=u+9*j:endswitch:                         endfunc
820 func e():                    f=r/129:m=-m:apage(1)
830  f=sqr(pow(abs(v-f),2)+pow(abs(u-r mod 129)*sqr(3),2))*5
840  g= f/10:             if f=g*10 then{}else{f=f+10}
850  f= f/10:          fill(244-128*m,0,267-128*m,15,0)
860 if mid$(right$("  "+itoa(f),3),2,1)=" " then{g=252}else{
870 if mid$(right$("  "+itoa(f),3),1,1)=" " then{g=248}else{
880  g=244}}:symbol(g-m*128,0,itoa(f),1,1,1,255,0):g=0:f= 1
890                    switch w:      case 0:f=-f:case 3:f=-f
900               for h=-7 to 7: for i=abs(h)-14 to 14-abs(h)
910    y(g)=x((  h  )*f+u+((     i    )*f+v)*129): g=g+1:i=i+1
920    next:next:           break:      case 1:f=-f:case 4:f=-f
930               for h= 0 to 7: for i=          0 to  7+    h
940    y(g)=x((  h-i)*f+u+((  h+i-14)*f+v)*129): g=g+1
950    next:next:for h= 0 to 6: for i=          0 to 13-    h
960    y(g)=x((7  -i)*f+u+((h+h+i- 5)*f+v)*129): g=g+1
970    next:next:           break:      case 2:f=-f:case 5:f=-f
980               for h= 0 to 7: for i=          0 to  7+    h
990    y(g)=x((7  -i)*f+u+((h+h-i- 7)*f+v)*129): g=g+1
1000   next:next:for h= 0 to 6: for i=          0 to 13-    h
1010   y(g)=x((6-h-i)*f+u+((  h-i+ 8)*f+v)*129): g=g+1
1020   next:next:    endswitch:     f=           1: g=  0
1030              for h=-7 to 7: for i=          0 to 14-abs(h)
1040 j=(6+f+h)/2*3+6-f-16*(m-1):                k=abs(h)+i+i+1
1050 bg_put(f,j,  k,8+256*y(g)):bg_put(f,1+j,  k,10+256*y(g))
1060 bg_put(f,j,1+k,9+256*y(g)):bg_put(f,1+j,1+k,11+256*y(g))
1070                              g=g+1:next:f=1-f:next:endfunc
1080 /* DIKE              BAS          6670  94-05-16  12:00:00
```

▶横内さんへ，「プロドライバーとバーチャレーシングで勝負」ですが，ジャン・アレジと石橋貴明が「生ダラ」というテレビ番組で対戦していました。アレジは結構うまかったですよ。
今井　佑(17)東京都

SIDE A

力学シミュレーションの約束ごとと高速定常円旋回運動

Tan Akihiko 丹 明彦

今月からいよいよ本格的に車の挙動をシミュレートしていく
ハンドルを回すと車が曲がっていく，という基本的な動作から
コーナリングにおける車の挙動をさらに突き詰めていく

「INDY CAR RACING」再び

先日，弟が遊びにきて，昨年紹介した「INDY CAR RACING」にはまっていった。彼は「WORLD CIRCUIT」を遊ぶためだけに私からAMIGAを買い取った過去をもっているほどの，3Dカーレースゲーム好きである。

「INDY CAR RACING」はPC互換機で動作するカーレースシミュレーションゲームで，現時点ではパーソナルコンピュータで遊べるカーレースシミュレーションの最高峰といっていい。「INDIANAPOLIS 500: the simulation」を制作したチームの最新作にふさわしくインディカーの挙動は精密を極め，テクスチャマッピングを施されたコースや車体は，特殊なグラフィックエンジンを搭載していないマシンの上で動いているとは思えないほど滑らかに動く。

さて，私はいままで，「INDY CAR RACING」はシミュレーションが精密すぎて，一般人ではきちんと車を走らせることができないと思い込み，運転する努力を放棄していたので，弟が10分もしないうちにラグナ・セカを周回し始めるのを見て大いに反省したのであった。インディといえばオーバルコースを連想するが，ラグナ・セカはいくつかのコーナーからなるロードコースで，コークスクリューと呼ばれる名物コーナーをもっていることで有名だ。

で，このコースの攻略がやたら燃えるらしい。アクセル全開でいけるコーナーもあれば，きちんと減速してアクセルコントロールしながら抜けていくコーナーもある。特に曲がりくねりながら急激に落ち込む下り坂コーナーであるコークスクリューは難所で，ブレーキングポイント，ライン取り，ハンドルの切れ角，アクセル開度，これらをすべて適切なタイミングでコントロールしないとコースアウトしたりスピンしたりする。限界付近での車の挙動を楽しむことができるのだ。

言葉ではいい尽くせないが，「INDY CAR RACING」は本当に偉大である。車の挙動を力学的にシミュレートして真剣に煮詰めていけば，いいものは作れるという証拠といってもいいだろう。もしもPC互換機をお持ちの方は，ぜひ遊んでみていただきたい。私が使っているのは486DX2/66MHzのマシンだから，それ以上のスペックだと問題ないだろう。音源にSound Blaster，操作系にアナログのアクセルとハンドルは必需品といっておこう。

力学シミュレーションの約束ごと

今月から本格的に車の力学的挙動に踏み込んでいく(ここまでで1年もかかってしまった)。ここでいくつかの約束ごとをしておく。

1) 当面は実数を多用する

ご存じのように，X68000の実数演算は決して速くない。が，力学シミュレーションは意外に厄介なもので，ほんのちょっとした要素が車の挙動を大幅に狂わせる。そうなった際に，原因が力学モデルの立て方にあるのか，それともコーディングミスにあるのか，はたまた数値誤差によるのか，といったことを特定しなくてはならない。ここで計算に整数を使っていると，精度不足，オーバーフロー/アンダーフローといった新たな要因が絡んでくる。固定小数点を使ってもこのへんの事情は変わらない。はっきりいって，余計な悩みを抱えたくないのだ。

というわけで，力学モデルを確立するまでは倍精度実数をベースにしてやっていく。必要な値のダイナミックレンジや精度が明らかになった点で，順次整数(または固定小数点)演算に移行することになるだろう。それまでは四則演算はもとより，三角関数や平方根にも実数演算を多用することにする。当面はX68030に数値演算コプロセッサを装着したシステムがないと悲惨なことになるだろう。

2) MKS単位系を用いる

これはそう取り決めるというだけの話。力学計算で用いる各物理量に対して,

長さにm(メートル)
質量にkg(キログラム)
時間にs(秒)

をそれぞれ単位として用いることにする。m, kg, sの頭文字を取ってMKS単位系と呼ばれる。たとえば1kmは1000[m]と表現され,時速36kmは10[m/s]と表現される。重力加速度は9.8[m/s²]である。

ちなみに角度は無次元量である。

C言語での単位変換テクニックとして,#defineを使うと比較的美しく書ける。たとえば,

```
#define HOUR        *3600.0
#define KM          *1000.0
#define KMPH   *(1.0 KM)/(1.0 HOUR)
```

と定義しておけば,Cコンパイラ(正確にはプリプロセッサも)がたとえば,

"2 HOUR"を"7200.0"[sec]に
"3.7 KM"を"3700.0"[m]に
"36 KMPH"を"10.0"[m/s]に

それぞれ変換してくれる。

逆に,m/s単位で算出された速度をkm/h表示に改めたい場合のために,次のような式も定義しておくと親切かもしれない。

```
#define _KMPH  *(1.0 HOUR)/(1.0 KM)
```

逆変換式の場合,アンダースコアを頭につけるという約束にしておこう。

なお,角度は無次元量で,計算上は単位がないが,日常的に用いる度[deg]と数学的に扱いやすいラジアン[rad]とのあいだでの変換式は必要であろう。MKS単位系ではラジアンを主に使うので,360°=2πという関係から,

```
#define DEG         *M_PI/180.0
```

という式を定義しておくとよい。

3) SLASHでの長さ1は1cmに相当する

これまたそう決めた。SLASHは仕様上ある程度以上の大きさの物体を扱えない(座標として最大16ビット,推奨12ビット程度が扱える)。対してサーキットは数kmのサイズがあるし,首都高速ともなると10km単位で考えなくてはならない。コース全体をひとつの物体で表現しようとすると,SLASHでの長さ1が1mほどになってしまい,とても微妙な車の動きなど表現できない。幸いSLASHではマップシステムが利用できるため,コースをブロック分けして近いブロックのみ表示するようにすれば,安全に精度を上げることも可能である。

MKS単位系からSLASHの長さへの変換は簡単で,

```
#define SLASHUNIT       *0.01
#define _SLASHUNIT      *100.0
```

を定義しておいて,座標変換パラメータの指定などで次のようにする。

```
parameter.x = (int)(x _SLASHUNIT);
parameter.y = (int)(y _SLASHUNIT);
parameter.z = (int)(z _SLASHUNIT);
```

この単位SLASHUNITは,実現しようとするモデルによって予告なく変更されることがある。

高速定常円旋回

今回はコーナリングについて少し進んだシミュレーションを行ってみる。

現時点で実現されているのは,極低速定常円旋回,つまり4輪がある1点を中心にして回るような円旋回運動である。タイヤはそれぞれが路面を転がっているだけで,一切横滑りしない(図1)。

今回は,より高速域でのコーナリングについて考える。車速が上がってくると,遠心力が発生して無視できない大きさになってくる。このため,タイヤから発生するコーナリングフォースを求めて力の釣り合いを考える必要が出てくるのである。

コーナリングの仕組み

車はなぜハンドルを切ると曲がるのか,ということは,あまり広く知られていない。教習所でも教えてはくれないようだ。

ハンドルを切れば確かにタイヤ(通常は前輪)は車体に対して斜めの方向を向く。しかしそれだけでは,車は曲がっていってはくれないのである。前輪が,その向いた方向に常に転がってくれるのなら,こんなに楽なことはない。しかし現実には,車はハンドルを切っただけ曲がっていくことはなく,それなりに速度を落とさないとカーブを曲がり切ることはできない。ここを理解していないと,峠道でオーバースピードでコーナーに突っ込み,アウトにふくらんでガードレールに接触した挙句に「この車アンダー

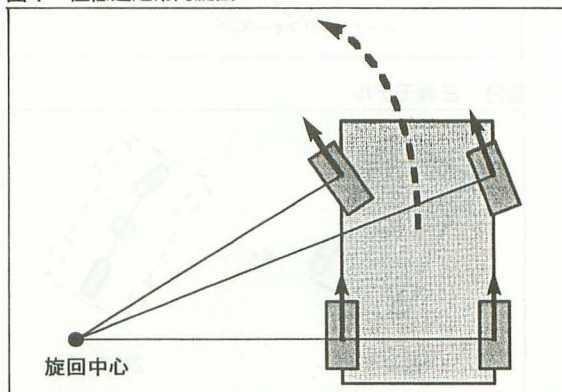
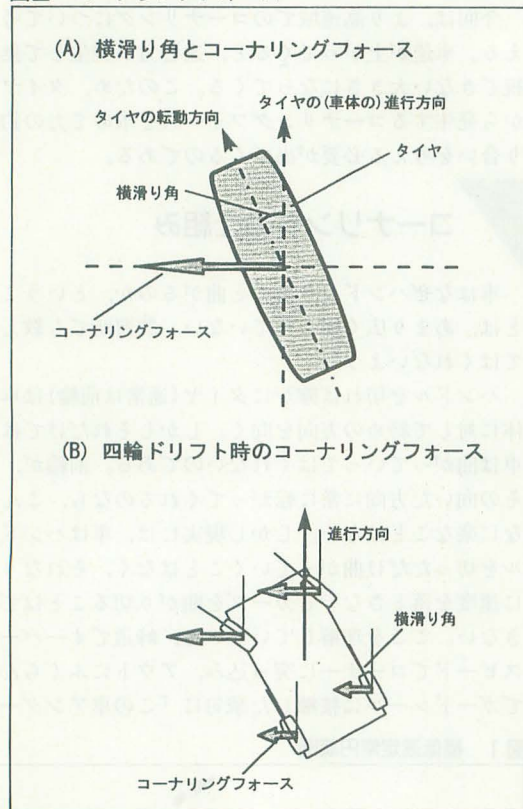
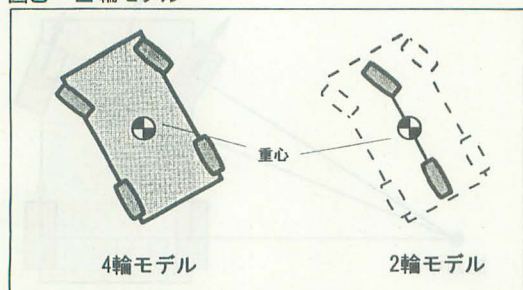



図1 極低速定常円旋回

だよ」などと寝ぼけたことをぬかすことになる。こういう手合いは，かの舘内端氏にいわせれば「お前がアンダー」ということになるのだ。

ちなみに「デイトナUSA」は，このへんを思い知らせてくれる希有のアーケードゲームだと私は思っている。上手な人は，どんな難所コーナーもなにごともなく回っていってしまうものだ。デイトナがアンダーだと思っている人は，自分のほうがアンダーなのだと反省して精進すること。

なお，アンダーとは「アンダーステア」の略で，定義はいくつかあるが，ドライバー側から見れば，ハンドルを切っただけ曲がっていかないように感じる状態のことを指す(あまり正確ではないが)。逆の状態は「オーバーステア」という。

話がそれてしまった。ハンドルを切ると車が曲がっていく理由の説明を続けよう。ハンドルを切ると，前輪(正確には「舵輪」なのであるが，ほとんどの車は前輪操舵である)は車の進行方向に対して斜めを向く。タイヤが転がる(自動車工学用語では「転動する」というようである)方向と実際に進んでいる方向との間に，角度がつくわけである。結果，タイヤは滑らかに転がり続けることができずに横滑りを起こす。この角度を横滑り角(スリップ角，スリップアングル)と呼ぶ。このときタイヤは接地点で捻れ，弾性変形する。ここでタイヤが復元しようとする力がコーナリングフォースとなる。コーナリングフォースは，タイヤの進行方向と垂直な方向に発生し，その大きさは(ある角度までは)横滑り角の増加につれて増大する(図2(A))。

もう少し正確にいうなら，コーナリングフォースは,タイヤの復元力(タイヤの転動方向と垂直な方向に発生する)のうち,タイヤの進行方向と垂直な成分によって発生する。

コーナリングフォースは後輪にも発生する。コーナリングの状況によっては，後輪にも横滑り角が生じるためである。この状況というのは，たとえば四輪ドリフトのように，車体そのものが進行方向とあからさまに違う方向を向いているような状況である(図2(B))。

コーナリングフォースのほかにもタイヤにはさまざまな力が関係するが，今回は話を簡単にするためにコーナリングフォースのみとした(こういういいわけを今後山ほどすることになると思うと気分が暗くなってしまう)。

図2 コーナリングフォース

図3 2輪モデル

高速定常円旋回運動

そこで高速定常円旋回運動である。車を発進させ，ある程度速度を上げた状態でハンドルを切る。そして挙動が安定した時点での力の釣り合いを調べてみよう。こんなシチュエーションは現実にはめったに起こり得ないのだが，いきなり過渡特性まで考えるのは荷が重すぎる。

さらに問題を単純にするために，2輪モデルを導入する(図3)。オートバイのモデルとは根本的に異なり，車はロール(横方向に傾く)しない。左右の荷重移動があったり，4輪が異なった種類の路面に接地したりといったことを考えないなら，このモデルである程度のことはわかる。

前輪(多少ながら後輪も)からはコーナリングフォースが発生する。定常旋回を仮定しているので，車体をひとつの質点と考え，前後のコーナリングフォースが合わせて車の重心に働くと考えることができる(もし定常旋回という仮定を外すと,回頭モーメントの発生を考える必要が出てくるので，車体を質点

でなく剛体と考えなくてはならない)。コーナリングフォースの合力は旋回運動を行うための求心力(遠心力と大きさが同じで向きが反対の力)として働くのである(図4)。

車速と車重，それにコーナリングフォースによる求心力から，円運動の公式，

$$遠心力(求心力)の大きさ=\frac{質量\times速度^2}{半径}$$

を用いて旋回半径を求めることができる。あとは車体を旋回中心(車の重心から求心力方向, 旋回半径の距離にある)のまわりに回転させればよい。

加速、減速も考えた高速定常円旋回運動

さらに駆動力と制動力を新たに加える。この運動まで「定常」かといわれると結構怪しいのだが，とりあえず影響は無視しよう。

駆動はエンジンからトランスミッションを介してタイヤから路面へ力を伝えることによって加速するための仕組みである。制動はブレーキパッドとブレーキローターの摩擦によって車の運動エネルギーを熱に変換する仕組みである。が，いずれもここではタイヤの転動方向に発生する前向き，または後向きの力として考える(このモデル化のまずい点は，制動力を後ろ向きの力ととらえている点である。このモデルでは車は停止することができず，ブレーキをかけすぎると後退を始めてしまうのである)。

高速定常円旋回運動の計算にあたっては，前後輪のコーナリングフォースにそれぞれ，駆動力と制動力の有効な成分(進行方向に垂直な成分)を加えて同様の計算を行えばよい(図5)。一般に制動力は前後輪に，駆動力はFF車なら前輪に，FR車なら後輪に発生する。4WD車なら両方だ。

この付近を突き詰めると，FF車はなぜコーナリング中に加速したほうが安定するのか，といったこともわかるのだが，まだまだそのへんに立ち入るには経験が足りない。過渡特性の挙動は奥が深いのだ。

終わりに

今回からは，高校から大学教養あたりまでの力学の知識と，それを上手に手抜きしながら使うテクニックが必要になっていく。もう私にとっては結構昔のことだし，勉強し直しに近いこともやっている。当時は目的意識もなく，漫然と教えられることを覚えていくという「お勉強状態」だったことを思い知らされる。単位はどうにか取れていても，本質を理解していないことが，こうした応用の場面になるとたちまち露呈するのだ。お勉強をなめちゃいけないぜという教訓を学生さんへの忠告としつつ，また次回お会いしよう。

［参考文献］
1) 「自動車工学」，樋口健治監修/自動車工学編集委員会編著，山海堂，ISBN4-381-10022-0，1988年
2) 「自動車力学」，景山克三/景山一郎共著，理工図書，ISBN4-8446-0356-6，1984年
3) 「スポーツカー　理論と設計」，コーリン・キャンベル著/篠田義明＋柏木二郎共訳，二玄社，ISBN4-544-04010-8，1971年

図4　高速定常円旋回

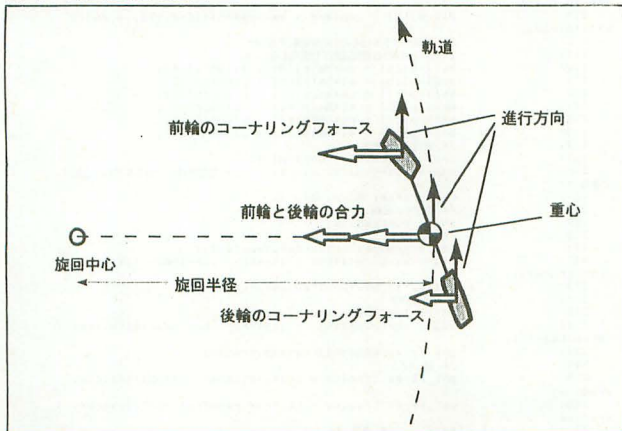
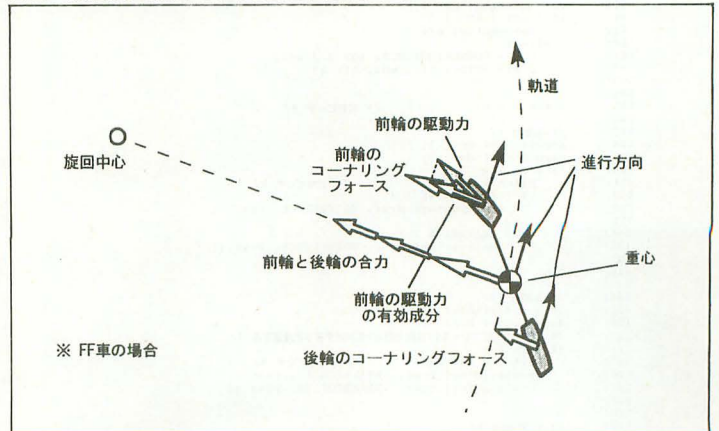



図5　駆動力を加味した高速定常円旋回（制動力も同様）

リスト（参考）

```
        1: /*
        2:  *   drive.c(参考)
        3:  *   - 車の動作
        4:  *     Aug. 1994  丹 明彦(Oh!X)
        5:  */
        6:
        7: /* ベクトル操作マクロ */
        8: #define dotproduct(V1,V2)          (V1[0]*V2[0]+V1[1]*V2[1]+V1[2]
*V2[2])
        9: #define scalevector(V1,S,V2)       {V1[0]=(S)*V2[0];V1[1]=(S)*V2[
1];V1[2]=(S)*V2[2];}
       10: #define addvector(V,V1,V2)         {V[0]=V1[0]+V2[0];V[1]=V1[1]+V
2[1];V[2]=V1[2]+V2[2];}
       11: #define subvector(V,V1,V2)         {V[0]=V1[0]-V2[0];V[1]=V1[1]-V
2[1];V[2]=V1[2]-V2[2];}
       12: #define length(V)                  sqrt(V[0]*V[0]+V[1]*V[1]+V[2]*
V[2])
       13: #define zerovector(V)              {V[0]=V[1]=V[2]=0.0;}
       14: #define crossproduct(V,V1,V2)      {V[0]=V1[1]*V2[2]-V1[2]*V2[1];
¥
       15:                                  V[1]=V1[2]*V2[0]-V1[0]*V2[2];
¥
       16:                                  V[2]=V1[0]*V2[1]-V1[1]*V2[0];
}
       17: #define copyvector(V1,V2)          {V1[0]=V2[0];V1[1]=V2[1];V1[2]
=V2[2];}
       18: #define dumpvector(N,V)            printf("%s=(%.2f,%.2f,%.2f)¥n"
,N,V[0],V[1],V[2])
```

```
 19: #define EPS                     (1.0e-5)
 20:
 21: VECTOR3 dir  = { 0.0, 0.0, 1.0 },    /* 車体進行方向 */
 22:         dirf = { 0.0, 0.0, 1.0 },    /* タイヤ進行方向前輪 */
 23:         dirr = { 0.0, 0.0, 1.0 };    /* タイヤ進行方向後輪 */
 24: VECTOR3 revf, revr;                  /* タイヤ回転方向前後輪 */
 25: double  scf, scr;                    /* スリップアングルの余弦前後輪 */
 26: VECTOR3 cff, cfr;                    /* コーナリングフォース前後輪 */
 27: VECTOR3 drvf, drvr;                  /* 駆動力前後輪 */
 28: VECTOR3 brkf, brkr;                  /* 制動力前後輪 */
 29: VECTOR3 tire, tiref, tirer;          /* タイヤの合力前後輪 */
 30: VECTOR3 sidedir;                     /* 車体進行方向に向かって右方向 */
 31: double  force;                       /* 力の大きさ */
 32: double  ve = 0.0;                    /* 速度 */
 33:
 34: /* 単位変換(試作) */
 35: #define HOUR     *3600.0                     /* 時間 */
 36: #define KM       *1000.0                     /* キロメートル */
 37: #define KMPH     *(1.0 KM)/(1.0 HOUR)        /* キロメートル毎時 */
 38: #define _KMPH    *(1.0 HOUR)/(1.0 KM)        /* キロメートル毎時にする */
 39: #define G        9.8                         /* 重力加速度 */
 40: #define KGF      *G                          /* キログラム重 */
 41: #define DEG      *(M_PI/180.0)               /* 度 */
 42: #define _DEG     *(180.0/M_PI)               /* 度にする */
 43:
 44: /* SLASHとの長さ変換 */
 45: #define SLASHUNIT   *0.01        /* SLASH単位 */
 46: #define _SLASHUNIT  *100.0       /* SLASH単位にする */
 47:
 48: /* 駆動形式 */
 49: #define FRONTDRIVE      0
 50: #define REARDRIVE       1
 51:
 52: /* 旋回運動に関する定数または近似式 */
 53: #define CORNERINGFORCE(C)   (20.0 KGF *acos(C)*C _DEG)
 54: #define WEIGHT              (1000.0)
 55: #define DRIVEFORCE          (500.0 KGF)
 56: #define BRAKEFORCEF         (1000.0 KGF)
 57: #define BRAKEFORCER         (500.0 KGF)
 58:
 59: void    drive()
 60: {
 61:     double  wd1, wd2;
 62:     VECTOR3 wv1;
 63:
 64:     /* 前回の時刻との差 */
 65:     t2 = ONTIME();
 66:     dt = TIMEDIFFERENCE(t2,t1)/100.0;
 67:     t1 = t2;
 68:
 69:     /* ステアリング角 */
 70:     theta = ITOD((128-x)*testcar.steeringratio);
 71:
 72:     /* タイヤ回転方向(実際にタイヤが向いている方向) */
 73:     si = sin( theta ); co = cos( theta );
 74:     revf[0] = -si*ba[p][0] + co*bc[p][0];
 75:     revf[1] = -si*ba[p][1] + co*bc[p][1];
 76:     revf[2] = -si*ba[p][2] + co*bc[p][2];
 77:
 78:     copyvector( revr, bc[p] );
 79:
 80:     /* スリップアングル余弦 */
 81:     scf = dotproduct( revf, dirf );
 82:     scr = dotproduct( revr, dirr );
 83:
 84:     /* タイヤ回転方向のタイヤ進行方向に垂直な成分 */
 85:     scalevector( wv1, scf, dirf );
 86:     subvector( cff, revf, wv1 );
 87:     scalevector( wv1, scr, dirr );
 88:     subvector( cfr, revr, wv1 );
 89:     /* コーナリングフォース */
 90:     wd1 = length( cff );
 91:     if ( wd1 < EPS ) {
 92:         zerovector( cff );
 93:     } else {
 94:         wd2 = CORNERINGFORCE( scf ) / wd1;
 95:         scalevector( cff, wd2, cff );
 96:     }
 97:     wd1 = length( cfr );
 98:     if ( wd1 < EPS ) {
 99:         zerovector( cfr );
100:     } else {
101:         wd2 = CORNERINGFORCE( scr ) / wd1;
102:         scalevector( cfr, wd2, cfr );
103:     }
104:
105:     pitch = 0.0;            /* にせピッチ */
106:
107:     /* 駆動力 */
108:     zerovector( drvf );
109:     zerovector( drvr );
110:     if ( rb ) {
111:         pitch = -PITCH;         /* にせピッチ */
112:         if ( FRONTDRIVE ) {
113:             scalevector( drvf, DRIVEFORCE, revf );
114:         }
115:         if ( REARDRIVE ) {
116:             scalevector( drvr, DRIVEFORCE, revr );
117:         }
118:     }
119:
120:     /* 制動力 */
121:     zerovector( brkf );
122:     zerovector( brkr );
123:     /* 手抜きなのでブレーキ(=後向きの力)をかけすぎると後退する */
124:     if ( lb ) {
125:         pitch = PITCH;          /* にせピッチ */
126:         scalevector( brkf, -BRAKEFORCEF, revf );
127:         scalevector( brkr, -BRAKEFORCER, revr );
128:     }
129:
130:     /* 前輪の合力 */
131:     addvector( tiref, drvf, brkf );
132:     addvector( tiref, tiref, cff );
133:     /* 後輪の合力 */
134:     addvector( tirer, drvr, brkr );
135:     addvector( tirer, tirer, cfr );
136:     /* 前後輪の合力 */
137:     addvector( tire, tiref,. tirer );
138:
139:     /* 進行方向に向かって右向きのベクトル */
140:     crossproduct( sidedir, bb[p], dir );
141:     /* (有効な)加速度 */
142:     wd1 = dotproduct( tire, dir );  /* 力の進行方向成分 */
143:     wd2 = wd1/WEIGHT;       /* 加速度 */
144:     /* 加速度に前回の時刻との差をかけて加えれば速度が出る */
145:     ve += dt*wd2;
146:     /* 前後輪の合力の大きさ=遠心力の大きさ */
147:     force = dotproduct( tire, sidedir );
148:     /* 旋回半径 */
149:     if ( force > EPS ) {
150:         radius = (WEIGHT)*ve*ve/force;
151:     } else if ( force < -EPS ) {
152:         radius = -(WEIGHT)*ve*ve/force;
153:     } else {
154:         force = 0.0;
155:         radius = 0.0;
156:     }
157:     /* 速度に前回の時刻との差をかければ移動量が出る */
158:     run = dt*ve;
159:
160:     if ( ve == 0.0 ) {
161:         roll = 0.0;         /* にせロール */
162:         copyvector(  b[1-p],  b[p] );
163:         copyvector( ba[1-p], ba[p] );
164:         copyvector( bb[1-p], bb[p] );
165:         copyvector( bc[1-p], bc[p] );
166:     } else {
167:         if ( force < -(10.0 KGF) ) {    /* 左折 */
168:             roll = -ROLL;   /* にせロール */
169:             centerx = b[p][0] - sidedir[0]*radius;  /* 旋回中心
*/
170:             centery = b[p][1] - sidedir[1]*radius;
171:             centerz = b[p][2] - sidedir[2]*radius;
172:             /* 旋回中心を中心として車体を回転する */
173:             co = radius / sqrt( radius*radius + run*run );
174:             si = run / sqrt( radius*radius + run*run );
175:             /* 位置ベクトルは旋回中心からの相対位置ベクトルを回転する */
176:             b[1-p][0] = centerx + radius*co*sidedir[0] + radiu
s*si*dir[0];
177:             b[1-p][1] = centery + radius*co*sidedir[1] + radiu
s*si*dir[1];
178:             b[1-p][2] = centerz + radius*co*sidedir[2] + radiu
s*si*dir[2];
179:             /* 基底ベクトルはそのものを回転する */
180:             /* bb(β軸)は旋回運動では変化しない */
181:             ba[1-p][0] =  co*ba[p][0] + si*bc[p][0];
182:             bc[1-p][0] = -si*ba[p][0] + co*bc[p][0];
183:             ba[1-p][1] =  co*ba[p][1] + si*bc[p][1];
184:             bc[1-p][1] = -si*ba[p][1] + co*bc[p][1];
185:             ba[1-p][2] =  co*ba[p][2] + si*bc[p][2];
186:             bc[1-p][2] = -si*ba[p][2] + co*bc[p][2];
187:             /* 進行方向の更新 */
188:             copyvector( dir, bc[1-p] );
189:             copyvector( dirf, dir );    /* 旋回半径が十分大きければ近似
できる */
190:             copyvector( dirr, dir );
191:             /* タイヤの転動 */
192:             phi = run/radius;
193:             /* 前左 */
194:             wd1 = (radius - testcar.fshaft/2);
195:             wd2 = sqrt( wd1*wd1 + (testcar.wheelbase)*(testcar
.wheelbase) );
196:             phi_fl += wd2*phi/testcar.fradius;
197:             /* 前右 */
198:             wd1 = (radius + testcar.fshaft/2);
199:             wd2 = sqrt( wd1*wd1 + (testcar.wheelbase)*(testcar
.wheelbase) );
200:             phi_fr += wd2*phi/testcar.fradius;
201:             /* 後 */
202:             phi_rl += (radius - testcar.rshaft/2)*phi/testcar.
rradius;
203:             phi_rr += (radius + testcar.rshaft/2)*phi/testcar.
rradius;
204:         } else if ( force > (10.0 KGF) ) {  /* 右折 */
205:             roll = ROLL;    /* にせロール */
206:             centerx = b[p][0] + sidedir[0]*radius;  /* 旋回中心
*/
207:             centery = b[p][1] + sidedir[1]*radius;
208:             centerz = b[p][2] + sidedir[2]*radius;
209:             /* 旋回中心を中心として車体を回転する */
210:             co = radius / sqrt( radius*radius + run*run );
211:             si = run / sqrt( radius*radius + run*run );
212:             /* 位置ベクトルは旋回中心からの相対位置ベクトルを回転する */
213:             b[1-p][0] = centerx - radius*co*sidedir[0] + radiu
s*si*dir[0];
214:             b[1-p][1] = centery - radius*co*sidedir[1] + radiu
s*si*dir[1];
215:             b[1-p][2] = centerz - radius*co*sidedir[2] + radiu
s*si*dir[2];
216:             /* 基底ベクトルはそのものを回転する */
217:             /* bb(β軸)は旋回運動では変化しない */
218:             ba[1-p][0] = co*ba[p][0] - si*bc[p][0];
219:             bc[1-p][0] = si*ba[p][0] + co*bc[p][0];
220:             ba[1-p][1] = co*ba[p][1] - si*bc[p][1];
221:             bc[1-p][1] = si*ba[p][1] + co*bc[p][1];
222:             ba[1-p][2] = co*ba[p][2] - si*bc[p][2];
223:             bc[1-p][2] = si*ba[p][2] + co*bc[p][2];
224:             /* 進行方向の更新 */
225:             copyvector( dir, bc[1-p] );
226:             copyvector( dirf, dir );    /* 旋回半径が十分大きければ近似
できる */
227:             copyvector( dirr, dir );
228:             /* タイヤの転動 */
229:             phi = run/radius;
230:             /* 前左 */
231:             wd1 = (radius + testcar.fshaft/2);
232:             wd2 = sqrt( wd1*wd1 + (testcar.wheelbase)*(testcar
.wheelbase) );
233:             phi_fl += wd2*phi/testcar.fradius;
234:             /* 前右 */
235:             wd1 = (radius - testcar.fshaft/2);
236:             wd2 = sqrt( wd1*wd1 + (testcar.wheelbase)*(testcar
.wheelbase) );
237:             phi_fr += wd2*phi/testcar.fradius;
238:             /* 後 */
239:             phi_rl += (radius + testcar.rshaft/2)*phi/testcar.
rradius;
240:             phi_rr += (radius - testcar.rshaft/2)*phi/testcar.
rradius;
241:         } else {    /* 直進 */
242:             roll = 0.0; /* にせロール */
243:             b[1-p][0] = b[p][0] + dir[0] * run;
244:             b[1-p][1] = b[p][1] + dir[1] * run;
245:             b[1-p][2] = b[p][2] + dir[2] * run;
246:             copyvector( ba[1-p], ba[p] );
247:             copyvector( bb[1-p], bb[p] );
248:             copyvector( bc[1-p], bc[p] );
249:             /* タイヤの転動 */
250:             wd1 = run/testcar.fradius;
251:             phi_fl += wd1;
252:             phi_fr += wd1;
253:             wd1 = run/testcar.rradius;
254:             phi_rl += wd1;
255:             phi_rr += wd1;
256:         }
257:     }
258:     return;
259: }
```

SIDE B

SIDE B
骨休めに3Dゲームを考察する

Yokouchi Takeshi 横内 威至

今回はちょっと趣向を変えて，SLASHの次期バージョンの話
そして，アーケードゲームとして登場している3Dポリゴンゲームを考察してみる
我々は3Dゲームになにを求めているのか，しばし考えてみよう

7月23日，期待していた十勝24時間耐久レースを観戦してきた。GT選手権のスペシャルステージということで，ここの成績がポイントに一切関係しないため，エントリーする車があまりいないのでは，と怪しんでいたのだが，結局45台もの車がエントリーしていた。

しかし車種に結構偏りがあったのは残念。クラスによっていろいろあったが，GT-Rとシビックがエラく多い。しかたないのだが，ル・マン並みのバラエティがほしかった。せめてフェラーリだとかスープラ，アンフィニRX-7，コマーシャルでは姿があったシルビアなんかも走っていてほしかった。そんななかでも，渋い車としてはMR2と旧RX-7がいた。特に旧RX-7のエンジン音が妙に印象に残る。ロータリーだよな，あれは。さらに上位2台のGT-RとNSXとタメ張る音を鳴らしてイキっていたのがたまらない。脳髄を直撃して全身をシビれさせるエンジン音は，生でなければ味わえないものだ。

そして1300C.C.クラスのカルタス。もちろんトップクラスのGT-RやNSXと比べてはならない。だがカルタスも負けてはいない。明らかにハードチューンされたエンジンが感覚を麻痺させるような強烈な音をたて，コーナーを限界近いスピードでクリアしていく姿は勇ましい。走る車はカルタスでも十分にカッコイイのだ。俺が24耐で最も好きなのは夕暮れどきと朝焼けどき。赤く染まった空気の中を疾走する車はとてもドラマチックで泣ける。

応援したのはゼッケン23,プリンス東京フジツボGT-R。ちょうど我々の目の前で必ずバックファイアを噴いてくれるのを，勝手にファンサービスととらえて驚喜していたのがいまとなっては愚かしい。我々と同じく夜間はしっかりと寝ていたのにも好感がもてた(実際はデフトラブルらしいけど)。レースはトラブルでのピットインが響いて残念な結果に終わった。我々の応援がいたらなかったせいであると反省している。結局トップはGT-R。当然最上位のクラスにエントリーしていた2WDのユニシアジェックスGT-Rであった。アドバンNSXはドライブシャフトの破損が響いて2位に終わる。

ル・マンほどの華やかさはないものの，フジテレビのバックアップもあり，イベントなどかなりがんばっていた。しかし，まだまだ発展途上のレースだな，と思わせるふしもある。コースがドライバーから不満が出るほどの単調さだし，しかもサーキットのうちの半分しか使っていないのが疑問であった。

またデータ取りの不備，メモリオーバーで途中から順位が発表されなくなったのもなかなかの不手際。インフォメーションも少なすぎ。トップ6のみの情報ではほかの車を応援している人間には非常につまらない。もっとまともな表示板を設置して全車の周回数も表示してほしかった。これから回を重ねるごとにいろいろと見直されていくとは思うが。

もともと耐久レースはいろんな楽しみがあるし，なによりF1などとは違って，身近な車が熱いバトルを繰り広げるのが魅力だ。24時間耐え抜いたあとに残るなんともいえない安堵感。そして疲労感。やはり一度は最高峰のル・マンを観たいと感じた。十勝24時間耐久も，いずれはル・マンのように世界的にも注目されるレースになることを期待している。

SLASH ver.3.0

唐突にSLASH ver.3.0について書いておく。次のディスク(もみじ狩りPRO-68Kの次)に間に合うかはわからないが，まあそれはそれということに。追加される部分がいろいろとある。順を追って説明しようと思う。

球体の追加

元祖「スタークルーザー」でマメに使われていたのがこの球体。現在のところ有効な使いどころがわ

からないのだが，いい加減な爆発なんかにはそれなりの効果があるだろう。

そして「スタークルーザー」のように惑星の表示にも，と思うのだが，致命的な問題がある。この球は中心座標を指定し，半径を与えて円を描くだけなので球体といっても平面に等しい。ということでシェーディングに対応させることができないのである。ほかのポリゴンがシェーディングしているのに球だけはベタ塗りで妙に浮いてしまうのが困る。惑星などは，陰影がないと見るからに嘘臭いときも多いのであまり応用したくないのだ。また，SLASH制作開始時はあまり気にならなかった惑星のリング。前後判定がないので惑星の裏にまわっても見えてしまうのは心苦しい手抜きだ。

つまりほかの物体との相性がいまいちとなっているのである。解決方法はない，といったら終わりなので少しは考えている。陰影の段階をつけるのも手だが，現実的なレベルでは白か黒か，きっぱりと明暗を分けるのがやっと。ないよりはマシだろうから

図1　座標変換の無駄

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

X＝ax＋by＋cz
Y＝dx＋ey＋fz
Z＝gx＋hy＋iz

たとえば2点P1，P2があり，X座標だけが同じならば上のax，dx，gxは二度も同じ計算をすることになり，無駄が生じる

図2　無駄な計算の回避方法

無駄な掛け算を切り捨てるために，まずポイントリストを変更。図1のようにP1(x，y1，z1)，P2(x，y2，z2)があったら

```
X座標リスト
  1:x
Y座標リスト
  1:y1
  2:y2
Z座標リスト
  1:z1
  2:z2
```

のように分割してデータをもつようにする。これを基底座標リストとする。ポイントリストは，

```
P1(1,1,1)
P2(1,2,2)
```

として基底座標リストの番号を指定する

基底座標リストは行列，

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

で次のように計算される

```
X座標変換リスト
  1:ax,dx,gx
Y座標変換リスト
  1:by1,ey1,hy1
  2:by1,ey2,hy2
Z座標変換リスト
  1:cz1,fz1,iz1
  2:cz1,fz2,iz2
```

ここまでのデータが揃えば，あとは足し算だけで座標が求められる

この方法を採用する可能性もある。でもこれはこれでかなり厄介なアルゴリズムを導入しなくてはならない。さあどうしよう。

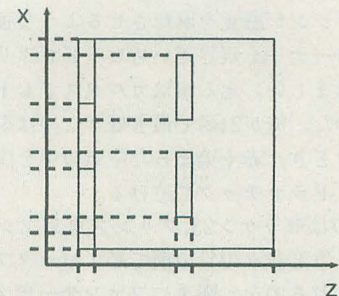
座標の分割

座標の変換の部分だが，まったく同じ座標さえなければ無駄はない，と考えられる。でも実際はまだ無駄がある。それはどういうことかといえば図1。このとおり，座標を構成するXYZのどれかの座標が同じものは無駄に掛け算を行っていることになる。やはり掛け算は重い処理なので避けるのが理想的。ということで図2のような概念で変更を行う。これによって計算部分は大幅なスピードアップが可能。スタンダードテストモデルのテスタロッサで25%もの高速化に成功した。

この方法が有効なものとして，サーキットの道路などがある。左右の紅白の縁，道路の白線，直線ではそうとうの効果が期待できる。また，非常に重いタイヤとホイールもかなりの省略が可能となる（図3）。

具体的にどのような変更が行われるかというと，座標リストに大幅なリニューアルが施される。いまいちつかみにくいデータになってしまったのでモデリングに手痛いキックバック。しかし大丈夫，いままでのデータからフィルタで一発変換してくれるようにする予定だ。まあそれほど難しいものではないから当然だな。

また，ここの部分からよりフレキシブルな座標系にしたいため，いままでのよう物体座標系の区別をなくして統一した。よって3D変換パラメータの指定も変更されている。すでに変換の基本は行列で与えることになっている。

図3　サーキットの道路の計算省略方法

たとえばこのように水平な道路のデータがあるとする。頂点の数は20個あり，本来ならば1点につき9回の掛け算が必要なので最終的には180回の掛け算を行わなくてはならない。しかし，Z座標は9種類の数値，X座標は5種類，Y座標は1種類だけである。1パラメータにつき3回の掛け算を行うため，全体で，
(9＋5＋1)×3＝45回
ですむ。これだけで4倍もの高速化ができる

ドライバ追加

未定だが，ドライババージョンを追加するかもしれない。非常にメモリを食うドライバだ。ドライバにする以上，エラーフリーなんていうシステムは許されない。エラーチェックの追加，エラーコードの設定，そのほかいろいろと綿密に練らねばならない部分がかなりあると思う。ドライバにすることによる利点は，俺にはあまりわからない。まあ，遅くはなるけど扱いやすくなるのは確かかな？　あとはタイトルが表示されたり，アプリケーションの配布がやりやすくなったり（システムがなくなるだけでそうとう減量できるはず）といたれりつくせり。ドライバってのも楽しそうじゃないか。これでユーザーの幅でも広がればいいのだが。

そのほか

そのほか，時間があればソースをもう一度見直してリストの視認性を上げたりとチョロっといじっている。しかし，あまりに膨大になっているので効率が格段に下がっているのが現状。またエラーチェックもしっかりと行われていない部分もあり，バグが少しずつ出始めたりとハマってきているのも事実だ。もう少し煮詰め直していらないコールを削ったりもしたいのだが，このあたりは非常にやりにくい状況となっている。

あとは各所のアルゴリズムの再検討，特にシェーディングについてはもっとしっかりとしたアルゴリズムが存在しているので，そのくらいのことはやらねばならないだろう。

シミュレータに向けて

今月は都合により，申しわけないがちょっと技術的なこととは離れたことをやっている。近頃3D関係の様子が急速に進化してきているので，ここで一度どんな感じかを探ってみようと思う。

●ゲームメーカーの実験開発期

3Dが巷に溢れてきたのはもう7,8年も前だろうか。「ウイニングラン」（ナムコ）がアーケード初のポリゴンゲームだったと記憶している。なにも知らない者は「とにかくヘボい」としかいわなかったものだ。そのちょっと前には「アウトラン」なんかが出回っていた時代だったからしかたないだろう。ゲームとしては難易度が高く，一般人にはかなり厳しいレベルだったようだ。また，理論的な部分もまだ完成してはないだろうし，攻略法があるあたりもシミュレータらしくない。処理速度も遅く，いま見ればかなり殺風景に感じるはずである。まあ，アーケード初のポリゴンゲーム，ということで新しい時代の先駆者であったことは間違いない。

次に思い出すのは「トップランニング」（タイトー）だ。「ミッドナイトランニング」の続編だが，こちらは完全にポリゴンシミュレータ。着陸だけを行う奇妙なフライトシミュレータである。風速の影響が強烈で難しい。これもいまから考えればやはり処理が重く，操作のレスポンスがあまりよくはなかったのを覚えている。とはいっても，ジャンボ機を操作したことがないので，ひょっとしたら実際の操作感覚と同じなのかもしれない。色使いもセンスよく，リアルなムードがたまらなかった。

次は「エアインフェルノ」（タイトー）。大型タンカー，ビル街，火山，砂漠でのヘリコプターの消火，救助活動を行う渋いネタであった。ヘリコプターの操作といっても，ゲームらしく非常に扱いやすいものであった。処理はまだ遅かったが，ゲームとしては面白いものだった。

そのあとは「ウイニングラン'91」（ナムコ）があったが，「ウイニングラン」とほぼ同じ。ナムコのドライビングシミュレータは，その後「リッジレーサー」までは似たようなものであった。

●完成された技術

そんな中でブレークスルーとなったのが「バーチャレーシング」（セガ）。秒間60コマの画期的な高速処理，しかも緻密にモデリングされた綺麗なコース，ダイナミックな視点など，3Dの優れた部分を見せつけた現代の3Dモノのスタンダードとなっている。このあたりからは学ぶべき点は多い。

まずは車のリアルな挙動。タイヤのグリップがモロに現れるあたりにはかなり驚いたものである。しかし，これは正確に物理的な法則をシミュレートしているようには思えない。これは勝手な予想であるが，まず車体の方向に段階がついているように思う。ストレートではしっかりと平行して直進できるあたりも嘘臭い。

また，限界挙動に関しては明らかに嘘を感じる。重心を中心とした単純な回転によって表現されるテールスライドなんかも，どうもリアルに感じられない。テールスライドをしながら縁石に乗り上げたときに，立て直そうとカウンターを当てたりしても，すでにスピンモードに入っているのはやはりゲームの領域であろう。

そのほかにおいてはきわめて優秀。アウト・イン・アウトのライン取りを正確に決めることによって高速，かつ美しい走りができるのはかなり快感。やり込んだゲームがこれだけなのでなんともいえないが，ドライビングで熱くなれたのはこのゲームだけ。また，グランプリモード（周回数20周）がゲームとし

ての出来を格段に高めている。1秒を切り詰めて周回を重ね続けるバトル。一瞬のミスが致命的となるのだ。技術が必要なのは当然だが，皆のレベルが近づくともう精神力の勝負。トップに追いつこうとあせった奴，トップで「ミスったら終わり」というプレッシャーに負けてミスをする奴，これこそがレースの勝負だ。シミュレーションゲームとしては，現在でも最高のレベルにあるものだと思う。あえて文句をいえば，対戦のときに登場するコンピュータカーは本当に無意味。素直に勝負させてもらいたい。もしくはプレイヤーごとにコンピュータカーがバラバラなのをなんとかすべきである。

ゲーム以外でも，サーキットを走る車を大胆な視点でカメラが追うデモや視点切り替えなどは革命と呼べる演出だった。ポリゴンは本当にイイな，と思わせてくれたゲームだ。

次に面白かったのは「ハードドライビン」，「レースドライビン」(アタリ)。処理速度の遅さが致命的だったが，ゲームとしては面白い。車でジェットコースターもどきの道路を走り抜けるという，非常に興味深い内容。山肌であろうとポリゴン上をどこでも走れるのは，かなり複雑な演算をこなしているとみえる。うまく使えば山肌に片輪を乗り上げ，ジャンプして一回転して着地なんてことも可能。インスタントリプレイで派手なクラッシュシーンを見られるのも嬉しいかぎりだ。シミュレータというより，ゲームとしては面白いものだったと思う。だがこのゲームの有名な点は，「本日のチャンピィヨン」，「コインいっこ入れる」などのイッてる和訳。カルチャーショックとはまさにこのことだ。

あとは「スターブレード」(ナムコ)。ドライブゲームではなくて「スターウォーズ」っぽいゲーム。アクロバティックな動きと魅せる映像。難易度以外は素晴らしい内容だと思う。

●新世代の映像技術

こうして3Dポリゴンが一般的になるといよいよ映像面の強化が課題となってくる。ポリゴンの無機質な映像にリアリティをもたせるのはやはりテクスチャマッピングだ。この技術をいち早く世に送り出したのが，「リッジレーサー」(ナムコ)である。いまでは十分有名だし，本誌でのインタビュー記事もあったので，読者の皆さんも知っていることだろう。映像に関しては色使いのセンスといい，文句なく美しい。視点切り替えなどの演出がセガの独占状態なので，そのへんの見せ方はやや不満が残る。

また，操作性は異常ともいえるドリフトコントロールが際立つ。トラクションをかければ速効テールスライド。そのままカウンターを当てなければ車体は90度以上も回転し，「逆走」状態でスピンもせずにコーナーを疾走する。きちんとカウンターを当てればドリフトし，立ち直るのもスムースにできる。コーナリング中のステアも凄い。右に左に，なんの抵抗もなく敏速に車が走る。現代を超越したタイヤグリップはわざとなのだろうか？　デモも明らかにドリフトを意識している。あらゆる車がテールを流してコーナーをクリアしていくのがカッコイイ。レースというより，ドリフトコンテストゲームと解釈できるくらいだ。

それに対し，本格的なドリフトが楽しめるのは「デイトナUSA」(セガ)。テールをスライドさせると，カウンターを当てなければスピン。アクセルとステアで微妙に調整し，立ち直りにも繊細な操作を必要とする点は実車の感覚にかぎりなく近い。コーナー出口よりもかなり早めに立て直しを始めなければフェイントがかかった挙動で直線につながらず，何度もカウンターを当てながら左右にケツを振ってしまい，最終的にあせってミスして壁に激突する。これは現実に起こり得る事故そのまま。恐ろしい。

演出もさすがにセガは抜きんでている。グラフィックの質感がアウトランのレベルでガサツというのが個人的に納得できないものの，書き込みの緻密さ，舞台の設定は芸術的。通常背景が映り込むガラスには雲が映り込む。見ているだけで楽しめる映像だ。サターンに移植される予定があるようだし，そうなったら家庭用として当然リプレイ機能により素晴らしい映像編集をしてくれるだろう。

このほかにも，3Dポリゴンを用いたアーケードゲームに「ポリネットウォーズ」(コナミ)や格闘シミュレータともいわれる「バーチャファイター」(セガ)などもあるが，個人的に守備範囲外な分野なので(格闘ゲームはやはり取り込み画像がいちばん)詳しい説明は割愛させていただく。

まとめ

最近，格段の進歩があったアーケードの3Dゲーム。X68000で実行しようとすると技術的に無理なものがほとんどだが，お手本とすべきもの，見習うべきものはしっかりとわかっている。車の挙動については「バーチャレーシング」「リッジレーサー」「デイトナUSA」と完成度の高さが目立つ。演出も当然そうだ。どこからなにを取り入れるか，バランスも大事だ。秒間10コマで「デイトナUSA」をやってもまったく意味がないだろう。ドリフトは微妙な挙動にこちらが反応できなければ制御不可能だ。

ということで今月はアーケードから学ぶ点をしっかりと意識しておきたい。突然こんな話になってしまったのは，プログラムのバグがとれず，やろうと思っていたことができなかったのだ。申しわけない。それでは来月。

THE SENTINEL

〈対応機種一覧〉●MZ-80 K/C/700/1500 ●MZ-80 B/2000 ●MZ-2500/2861 ●X1 ●X1 turbo/Z ●PC-8001/8801/88 ●SMC-777/C ●PASOPIA/5 ●PASOPIA/7 ●FM-7/77/AV ●MSX/2/2+/turbo R ●PC-286/386/486/9801/98/9821 ●X 68000/X 68030
掲載されたプログラムの利用には各機種用のS-OS "SWORD" システムが必要です。

第149部　シューティングゲーム作成講座(3)

第150部　怪しいZ80の使い方(未定義命令編)

●本当にMOOK化は可能なのか？

今回はいつもやっているTHE SENTINELの解説をお休みして，かねてより公言してきたS-OS "SWORD" MOOK化計画についてお話します。

まず，今回制作しようとしているものは，いままで発表されたすべてのシステムとアプリケーションを収録する，といった大規模なものにはなりません。できるかぎりのアプリケーションを収録したいのですが，あまり大きく動けない現状があります。

それは，S-OSユーザー数の減少です。要するにある程度の販売見込みがないと，ハデなことができないのです（S-OS "SWORD" MOOKのために編集人員もさけないし，スタッフも足りない)。

そこで，人数が少ないんだったらユーザー同士が協力して，S-OSユーザー自らの手でいままでS-OSが行ってきた功績を残していったらいいんじゃないか，と考えたのです。

確かに絶対数は少ないにしても，現在でも8ビット機，S-OS "SWORD" を利用しているユーザーがいることは確かなのです。そういった人たちとOh!Xが協力すれば，MOOKを世に送り出すことも不可能ではありません。

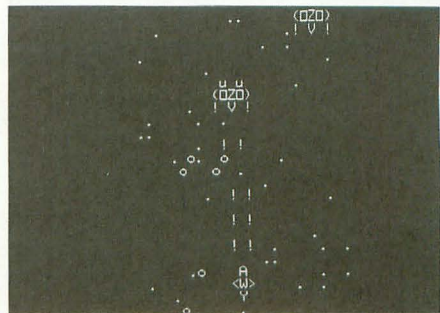
そして，ある程度作業が進み完成のめどがたった段階で購入希望者を募り，必要なだけ制作して読者の皆さんに配布しようと思っています。まだ，どういった形態にするかは決まっていませんが（応募者数によってどうにでも変わるので)，やれるかぎりのことをやって満足できるものを作りたいものですね。

●MOOK化計画，次なるステップへ

以上のように，S-OS "SWORD" MOOK化計画も方針が見え始めてきました。

そこで，「アプリケーションフリーソフト化計画」によって集められたものを中心にして，次のステップへ進もうと思っています。

次なるステップでなにをやるかというと，アプリケーションのマニュアル作成です。以前にも説明しましたが，プログラムについては「アプリケーションフリーソフト化計画」によって，配布に際しての問題がなくなりました。しかし，マニュアルだけは新たに書き起こす必要があるのです（著作権がらみ＋古い記事のファイルがないという状況から)。

そこで，THE SENTINELでは「マニュアル作成スタッフ」を募集します。官製ハガキ（官製ハガキのほうが嬉しいけどアンケートハガキでも可）に住所，氏名，年齢，電話番号を明記のうえ，

Oh!X編集部　THE SENTINEL
「マニュアル作成スタッフ募集」係

までお送りください。折り返し，具体的な作業内容を封書でお送りします。作業内容といってもマニュアルの打ち込みだけですので，遠隔地の方でも大丈夫です。気軽に応募してください。

また，時間的に余裕があるから協力できそうだけど，古いOh!X（MZ）なんて持っていないし……という方には，該当するマニュアルのコピーも一緒に同封します。この場合は，自分の持っているバックナンバーを応募ハガキに明記してください。

なお，どのアプリケーションのマニュアルを作成してもらうかは，Oh!X編集部で割り振らせてもらいます。やりたいもの，開発言語をバキバキに使いこなして自信のある方は，その旨を明記してください。作業を割り振るときに考慮します。

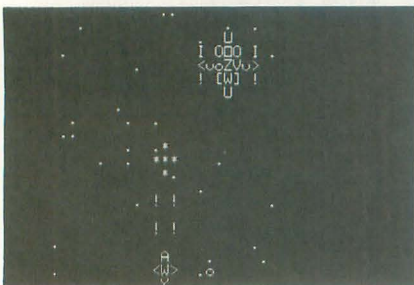
そして，打ち込んだマニュアルは，MS-DOSフォーマットのテキストファイルに収めてOh!X編集部まで郵送してください。メディアは5インチ2HDか3.5インチ2HD（3.5インチは2DDでもOK）が望ましいですね。編集作業上の都合ですが，ご協力をお願いします。

1994■インデックス

全機種共通
S-OS"SWORD"要

シューティングゲーム作成講座(3)

Uesugi Yuya
上杉 悠也

キャラクタープログラムの解説をとおして，ゲームについての基本的な構造を解説します。来月から本格的にゲームの制作に入りますので，これらの基礎知識をきちんと理解しておきましょう。

今月は個々のキャラクタープログラムの作り方を解説していきます。前回のワークなどの説明とともに非常に重要な部分ですので，じっくり腰を据えて（？）読んでください。

キャラクタープログラム制作上の注意点

ゲームのキャラクタープログラムを作るうえで，特に重要なことはリエントラント性の確保です。

リエントラントとは直訳すれば“再入力”のことで，簡単にいうと多重呼び出し（以後多重処理と呼ぶ）のことです。たとえば，Aのプログラムが実行を終える前に，さらにほかからAのプログラムが呼び出されるような状況が発生した場合でも，問題なく対応できるようにプログラムを組まなければなりません。

これは1回目で解説している複数キャラクターの呼び出しに関することで，ゲームを作るうえで最も基本的かつ重要な項目に属しています。

つまりAというザコキャラクターを複数個画面に出す場合，出現個数分のプログラムを用意するわけにはいきません（そんなことをしたら，膨大な数のプログラムを用意しなければいけなくなるので）。そこで複数のキャラクターを1つのプログラムで，同時に制御することになります（詳しくは1回目の基礎編を参照するとわかりやすいでしょう）。

そして，多重処理への対応の準備として，まず処理の細分化が必要です。たとえば，Aのキャラクターをa点からb点へ移動させる処理を行うとします。もし1番目のキャラクターでb点までの移動を最後まで行うとすると，そこで処理が停滞してしまい，ほかのキャラクターの処理が止まってしまいます。このような事態を防ぐために移動という処理を細分化して扱います。

具体的な手順としては，

1) 移動量の算出
2) 移動
3) 位置チェック
4) 到達していなければ2)へ

のように分割することができます。この手順で切るべき（ほかのキャラクターに処理を渡す）箇所は4)が最適です。

この処理で注意しなければいけないのは2)のところで，あくまでも1移動量分のみ移動したら，次の処理（この場合だと，3)の位置チェック）に実行を移さなければならないということです。そうしなければ，処理を細分化している意味がまったくないことになってしまいます。

処理の細分化で最も重要なところは，個々の処理で実行時間（処理時間の占有）をできるだけ短縮し，数多くの処理をこなす点にあります。

処理の細分化ができたところで次に多重処理への対応について説明します。多重処理へ対応していないプログラム（リスト1a）と，対応しているプログラム（リスト1b）の簡単なサンプルを示しておきます。

この2つの例を比較すれば気づくと思いますが，多重処理に対応していない（処理を手放さない）プログラムのほうが，対応しているプログラムよりも比較的単純に記述できます。それに対して，多重処理に対応させたプログラムは一見複雑そうな処理を記述しなければいけません（考えなければいけないことがより多くある）。

それではこの2つの違いについて考えてみましょう。

まず，リスト1aが単純な条件ループによって構成されているのに対して，リスト1bはまったくループを使用していません。これは，ループの代わりにシステムが順次呼び出しを行うためです。ループで処理を行う場合，その処理が終了するまでほかの処理に実行が移りません。それに対してシステムが順次呼び出しを行う場合は，いくらでもほかの処理を途中で実行することができます（当然といえば当然ですが）。

次に，ループで処理を構成している場合，キャラクターの表示やそのほかの煩雑な処理なども自前ですべて処理する必要がありますが，システムを利用する場合，それらの煩雑な処理はシステムが肩代わりしてくれます。

最後に，一度しか実行しない処理（初期化など）の扱い方の違いです。ループ処理ではリスト1aのようにループの外に処理を置くだけですが，システムにより呼び出される場合はひと工夫必要になります。リスト1bの場合はモードにより飛び先を変更する方法をとっています。この方法はYGCSでの標準的な分岐処理のやりかたで

す。そのほかの方法としては，フラグを設けて処理を飛ばして実行する方法などがあります（リスト1c)。

多重処理に対応させるには，もうひとつ重要なことがあります。それはワーク管理の問題です。

YGCSでは各キャラクターごとに32バイトのキャラクター管理ワークを確保しています。このうち上位24バイトはシステムで使用するためのワークですが，残り8バイトはユーザー用に開放されている汎用ワークです（前回のワーク構造の項目を参照してください)。キャラクターのプログラムを組む場合は，この32バイトのワーク以外は基本的には使用できません。

プログラム内に固定ワークを設けると，そのプログラムは多重処理に対応できなくなります。ちなみに固定ワークとは“DS”などで確保されたワークのことです。

これらすべての項目は，多重処理に対応させるための注意点です。個々のキャラクターのプログラムを組むときは，常に頭の片隅に置いておいてください。

メイン処理の解説

それでは実際にキャラクターの動く（まだ1体ですが）プログラム（リスト2）を例にしながら，YGCSの基本的な使い方を説明していきます。今回取り上げるのは，簡単な自機のプログラムです。大まかにシステムの動きも順に説明します。このリスト2は，以後の講座の基本（雛型）になるものですから必ず打ち込んでおきましょう。

まずはメインの処理を項目ごとに分けて説明します。

1：システムの初期化

= COLD_

ここでは画面の解像度の設定と（40桁モード)，ワークの初期化（0クリア）を行っています。ここでのワークはシステム内のワークのことで，ユーザー側のワークではありません。注意してください。

2：各種テーブルの設定

= CHRF_,GENF_,KEYF_

ここについては以前説明したので，今回は説明を省きます。詳しくはリファレンスマニュアルを参照してください。

3：仮想画面，システム変数の初期化

= HOT_

本来このコールは各ラウンドのスタート時に呼び出すコールです。

リスト3を見てください。このコールでまず初めに行っているのが，仮想画面の初期化です。仮想画面は全部で4面あって，0番の画面がVRAM転送用画面のため0クリアし（1回目は全画面書き換えを行うため)，そのほかが背景，敵，自機の各表示用画面でスペースでクリアしています。

次は仮想画面の内容をVRAMに転送し，実際のVRAMのクリアを行っているわけです（リスト3)。

このコールでは仮想画面の内容をチェックし，新たに描き換わったところのみVRAMに転送します。まず仮想画面1から3までで，最優先のキャラクターコード（スペースを除く）を取り出します。次にそのコードと仮想画面0番の内容とを比較して，同じであれば転送は行わず，違っていればVRAMへそのコードを転送します。そうすることにより，前回と変わらないコードに関しては転送を行わず，処理速度を稼いでいるわけです（毎回画面をまるごと転送していたのでは，遅くて使いものにならなくなるから)。がんばって，リストに目をとおしてみてください。内部処理を知っておくことは，決して無駄にはならないはずですから。

最後に行っているのは，ポーズフラグとシステムカウンタの0クリアと，キャラクター出現スケジュールテーブルのポインタの初期化です。

4：ユーザーワークの初期化

= MAIN_INIT

ここで行っているのは，プレイヤーフラグのクリアのみです。今回のサンプルでは，それしかワークを使用していません。

このフラグはプログラム全体から参照します。よって，固定変数として確保しておく必要があるわけです。

以上の処理が終わるとメインループに入ります。

5：強制終了チェック

= _BRKEY

これはS-OSのコールで，BREAKキーをチェックしています。もしBREAKキーが押されていれば終了処理に実行を移します。

6：一時停止チェック

= PUSC_

このコールはポーズフラグを反転しているだけです。もし結果（Zフラグ）が1であれば，なにもせずMAIN_LOOPへ戻ります。なお，この5，6の項目に関してはユーザーが自由に書き換えても結構です。特にほかへ影響が出るということはありません。

7：ゲーム処理

= MAIN_

ここからが本番。リスト4を見てください。これがMAIN_で行っている処理です。

まずキー入力の処理をまとめて行っていて，入力されたキーデータはシステム変数に保存されます。以後のキー情報を必要とする処理では，この変数を参照するようにしてください。これは情報の統一性を維持するためです。

具体的なキー入力の処理はリスト5を参照してください。KEY_READのほうが現在押されているキーの読み取りで，KEY_EDGEのほうは今回押された（重要）キーの読み取りを行うためのコールです。

基本的にキーの情報は2種類必要になります。それが上で説明したコールで読み取る，現在押されているキーと，今回押されたキーの情報です。ゲームを作るうえで，この2種類のキー情報は欠かせないものといえるでしょう。

さて，キー入力処理がすんだところで，次のCHR_GENでキャラクターの出現（発生）処理を行います（リスト6)。このコールは呼び出されるたびに，システムカウンタとユーザー定義のテーブルをチェックして，キャラクターの出現をコントロールするわけです。システムカウンタの値もここでカウントアップしています。

次のCHR_CTRLでは（リスト7)，上のCHR_GENによってワークに定義されたキャラクターを動かすわけですが，基本的にヒットチェックの呼び出しもここでしています。しかし，今回は関係ないので解説は省略することにします。今後必要になったときにまとめて解説する予定です。

このあとに行っていることは，キャラクターの表示関係の処理です（リスト8)。8行目のOBJ_CLRで仮想画面のクリアをし，45行目のOBJ_DISPで仮想画面にキャラクターを描いています。そしてその結果を

VRAM_TRANSでVRAMへ転送します。

これでMAIN_の説明はひととおり終わりました。要はこの処理を繰り返すことで，ゲームが動いているわけです。大まかな道筋でプログラムを見ていくと，実に簡単な構造であることに気づくと思います（初心者には難しいかもしれませんが)。自分でゲームのシステムを作る場合でも，このMAIN_と同じ処理を記述すればそれなりに作れるはずなので，機会があったらトライしてみるのもいいでしょう。といっても，この処理方法が最善であるとはいいません。ある程度自分でシステムを記述できるようになったら，各自でいろいろ工夫してみてください。

メイン処理の最後は終了処理です。が，今回は画面のクリアのみでS-OSへ制御を戻しています。

メイン処理の説明はここまでです。次はサンプルキャラクターの説明なのですが，基本的にキャラクターの作り方に関してはいままでの記事で説明してきました。よって今回特別に解説しなければならないことはありません。

さて，次回は？

今月はちょっと短いですが，ここまでです。次回からは，本格的にゲームそのものの制作に入ります。基本的なところで説明不足な点は多々あると思いますが，どうしても必要なことは，その都度取り上げる予定でいます。もし，それでもわからないようでしたらどんどん質問してください。それではまた来月（に載せられることを祈って)。

リスト1

```
 1: ; LIST 1a
 2: ;
 3: ;          実行が終了するまで、処理を手放さない例
 4: ;
 5: CHR_A:
 6:          LD      (IX+SPD),$10    ; Speed set
 7:          LD      H,TARGET_X      ; Target position set
 8:          LD      L,TARGET_Y
 9:          CALL    SINC_           ; 1:移動量の算出
10: CHR_A_LOOP:
11:          CALL    CHR_DISP        ; キャラの表示が必要
12:          CALL    MOVE_           ; 2:移動
13:          CALL    POS_CHK         ; 3:位置チェック
14:          JR      C,CHR_A_LOOP    ; 4:未到達なら2へ
15:
16:          LD      (IX),0          ; 自分を消す（終了）
17:
18:          RET
19:
20: 注）CHR_DISP POS_CHK は自前で用意する。
21:
22:
23: ; LIST 1b
24: ;
25: ;          処理を多重処理に対応させた例
26: ;
27: CHR_A:
28:          LD      HL,CHR_A_TBL
29:          JP      MODC_
30: CHR_A_TBL:
31:          DW      CHR_A_INIT
32:          DW      CHR_A_MOVE
33:
34: CHR_A_INIT:
35:          LD      (IX+SPD),$10    ; Speed set
36:          LD      H,TARGET_X      ; Target position set
37:          LD      L,TARGET_Y
38:          CALL    SINC_           ; 1:移動量の算出
39:
40:          LD      (IX+MOD),1
41:
42: CHR_A_MOVE:
43:          CALL    MOVE_               ; 2:移動
44:          CALL    POS_CHK             ; 3:位置チェック
45:          JR      NC,CHR_A_END        ; 4:到達なら終了
46:
47:          RET
48:
49: CHR_A_END:
50:          LD      (IX),0              ; 自分を消す
51:
52:          RET
53:
54: 注）POS_CHK は自前で用意する。
55:     表示はシステムに依存。
56:
57: ; LIST 1c
58: ;
59: ;          フラグによる分岐の例
60: ;
61: CHR_A:
62:          LD      A,(IX+24)
63:          OR      A
64:          JR      NZ,CHR_A_MAIN   ; Flag 分岐
65:
66:          INC     A
67:          LD      (IX+24),A       ; Flag set
68:
69:          LD      (IX+SPD),$10    ; Speed set
70:          LD      H,TARGET_X      ; Target position set
71:          LD      L,TARGET_Y
72:          CALL    SINC_           ; 1:移動量の算出
73: CHR_A_MAIN:
74:
75: 注）(IX+24)はユーザーワークで、キャラ発生時に
76:       初期化されているものとする。
```

リスト2

```
 1: ; LIST 2
 2: ; ==============================
 3: ;       SAMPLE PROGRAM
 4: ; ==============================
 5:
 6:         CSEG
 7:
 8:         INCLUDE SOSCALL.EQU
 9:         INCLUDE YGCSCALL.EQU
10:
11: ; ==============================
12: ;       PROGRAM MAIN
13: ; ==============================
14: START:
15:         LD      A,$C
16:         CALL    _PRINT          ; Text cls
17:
18:         CALL    COLD_           ; System init
19:         LD      HL,CHR_TBL
20:         CALL    CHRF_           ; CHR table address set
21:         LD      HL,CHR_GEN
22:         CALL    GENF_           ; CHR gene table address set
23:         LD      HL,KEY_TBL
24:         CALL    KEYF_           ; Key table address set
25:         CALL    HOT_            ; Screen init & etc...
26:
27:         CALL    MAIN_INIT       ; Main program init
28: MAIN_LOOP:
29:         CALL    _BRKEY          ; EXIT check
30:         JR      Z,EXIT
31:         CALL    PUSC_           ; Pause check
32:         JR      Z,MAIN_LOOP
33:
34:         CALL    MAIN_           ; Game main
35:
36:         LD      A,(PL_FLAG)
37:         OR      A               ; Stage clear check
38:         JP      NZ,EXIT
39:
40:         JR      MAIN_LOOP
41: EXIT:
42:         LD      A,$C
43:         CALL    _PRINT          ; Text cls
44:
45:         RET                     ; Return S-OS
46:
47: ; ==============================
48: ;           MAIN INIT
49: ; ==============================
50: MAIN_INIT:
51:         XOR     A
52:         LD      (PL_FLAG),A
53:
54:         RET
55:
56: ; ==============================
57: ;            PLAYER
58: ; ==============================
59: PLAYER:
60:         LD      HL,PLAYER_TBL
61:         JP      MODC_
62: PLAYER_TBL:
63:         DW      PL_MODE_0       ; 0 : Player init
64:         DW      PL_MODE_1       ; 1 : Player stage start
```

▶７月の木星と彗星の衝突すごかったですね。うちの小さな望遠鏡でも見えました。あんなのが地球で起こったらと思うとゾッとします。　　　吉永　克己(30)千葉県

```
 65:         DW      PL_MODE_2        ; 2 : Player main
 66:         DW      PL_MODE_3        ; 3 : Player stage clear
 67: ;
 68: ;       <<< PLAYER INIT
 69: ;
 70: PL_MODE_0:
 71:         LD      (IX),3           ; CHR size 3x3
 72:         LD      (IX+ATR),$01     ; Player mode
 73:         LD      HL,PAT_PL        ; CHR pattern set
 74:         LD      (IX+PAT),L
 75:         LD      (IX+PAT+1),H
 76:
 77:         LD      (IX+MOD),1       ; Next move mode
 78:
 79:         RET
 80: ;
 81: ;       <<< PLAYER STAGE START
 82: ;
 83: PL_MODE_1:
 84:         ;DEC    (IX+YPS)
 85:         LD      A,(IX+YPS)
 86:         SUB     1
 87:         LD      (IX+YPS),A
 88:         LD      A,(IX+YPS)
 89:         CP      20               ; Ypos = 20 ?
 90:         JR      NZ,PL_M1_RET
 91:
 92:         LD      (IX+MOD),2       ; Next move mode
 93: PL_M1_RET:
 94:         RET
 95: ;
 96: ;       <<< PLAYER MAIN
 97: ;
 98: PL_MODE_2:
 99:         CALL    PL_MOVE
100:
101:         CALL    KEYG_            ; Key read
102:         AND     $F0
103:         JR      Z,PL_M2_RET      ; Exit ?
104:
105:         LD      (IX+MOD),3       ; Next move mode
106: PL_M2_RET:
107:         RET
108: ;
109: ;       <<< PLAYER STAGE CLEAR
110: ;
111: PL_MODE_3:
112:         LD      A,(IX+YPS)
113:         SUB     2
114:         LD      (IX+YPS),A
115:
116:         CALL    OUTC_
117:         JP      C,PL_EXIT
118:
119:         RET
120: ;
121: ;       <<< PLAYER MOVE
122: ;
123: PL_MOVE:
124:         CALL    KEYG_            ; Key read
125:         AND     $F
126:         ADD     A,A
127:         LD      E,A
128:         LD      D,0
129:         LD      HL,PL_JTBL
130:         ADD     HL,DE
131:
132:         LD      A,(IX+XPS)
133:         ADD     A,(HL)           ; X move
134:         LD      (IX+XPS),A
135:         INC     HL
136:
137:         LD      A,(IX+YPS)
138:         ADD     A,(HL)           ; Y move
139:         LD      (IX+YPS),A
140:
141:         RET
142: ;
143: ;       <<<     PLAYER EXIT
144: ;
145: PL_EXIT:
146:         XOR     A
147:         LD      (IX),A
148:         INC     A
149:         LD      (PL_FLAG),A
150:
151:         RET
152: ;
153: ;       <<< MOVE DATA
154: ;
155: PL_JTBL:
156:         ;       dX_dY:dX_dY:dX_dY:dX_dY
157:         DB      00,00,00,-1,00,01,00,00
158:         DB      -1,00,-1,-1,-1,01,00,00
159:         DB      01,00,01,-1,01,01,00,00
160:         DB      00,00,00,00,00,00,00,00
161: ;
162: ;       <<< PATTERN DATA
163: ;
164: PAT_PL:
165:         DB      ' A '
166:         DB      '<W>'
167:         DB      ' Y '
168:
169: ; ==============================
170: ;       CHR PROGRAM TABLE
171: ; ==============================
172: CHR_TBL:
173:         ;       Label              ID   Name
174:         DW      PLAYER          ;   0 : Player
175:
176: ; ==============================
177: ;      CHR GENERATE TABLE
178: ; ==============================
179: CHR_GEN:
180:         ;       COUNT:COD:SRCHCNT:OFADR:XP:YP
181:         DW      $0000,$00+01*$100,$0000,17+25*$100 ; Player set
182:
183:         DW      $FFFF,$FF+00*$100,$0000,00+00*$100 ; Table end
184:
185: ; ==============================
186: ;          KEY TABLE
187: ; ==============================
188: KEY_TBL:
189:         DB      '89632147'       ; Direction key
190:         DB      'ZXCV'           ; Button key
191:
192: ; ==============================
193: ;         GLOBAL WORK
194: ; ==============================
195: PL_FLAG:        DS  1            ; Player flag 1/stage clear
196: ; ==============================
```

リスト3

```
  1: ; LIST 3
  2: ; ==============================
  3: ;   SYSTEM SCREEN & WORK INIT
  4: ; ------------------------------
  5: ; in    : non
  6: ; out   : non
  7: ; break : all
  8: ; ==============================
  9: HOT_INIT:
 10:         LD      A,$1
 11:         LD      B,0
 12:         CALL    VRAM_FILL        ; Vram0 fill
 13:         LD      A,$E
 14:         LD      B,' '
 15:         CALL    VRAM_FILL        ; Vram1-3 fill
 16:
 17:         CALL    VRAM_TRANS       ; VRAM display
 18:
 19:         LD      HL,0
 20:         LD      (PAUSE_FLAG),HL  ; Pause flag clear
 21:         LD      (SYS_COUNT),HL   ; System count clear
 22:         LD      HL,(GEN_TBL_ADR)
 23:         LD      (GENE_POS),HL    ; Generate table position reset
 24:
 25:         RET
 26:
```

リスト4

```
  1: ; LIST 4
  2: ; ==============================
  3: ;       SYSTEM JOB CONTROL
  4: ; ------------------------------
  5: ; in    : non
  6: ; out   : non
  7: ; break : all
  8: ; ==============================
  9: JOB_CTRL:
 10:         ; ***** KEY INPUT JOB
 11:         LD      A,(KEY)          ; Old key data get
 12:         LD      B,A
 13:         CALL    KEY_READ         ; Key read
 14:         LD      (KEY),A
 15:         CALL    KEY_EDGE         ; Key on edge check
 16:         LD      (KEY_ON),A
 17:
 18:         ; ***** MAIN CONTROL JOB
 19:         CALL    CHR_GEN          ; Character generate (count up)
 20:         CALL    CHR_CTRL         ; Character control
 21:         CALL    OBJ_CLR          ; Object clear
 22:         CALL    OBJ_DISP         ; Object display
 23:         CALL    VRAM_TRANS       ; VRAM display
 24:
 25:         RET
 26:
```

▶「世界最大の恐竜博」へ行ってきました。世界最大と豪語するのはちょっとうさん臭い気がしましたが，展示後半でのティラノサウルスの化石レリーフの美しさやマメンチサウルスの立体模型の大きさなどは圧巻でした。親指大から何十mまで，恐竜の世界って広いわ。　大嶋　靖浩(32)栃木県

リスト5

```
 1: ; LIST 5
 2: ; ==============================
 3: ;         JOY KEY READ
 4: ; ------------------------------
 5: ; in    : non
 6: ; out   : A = Key data
 7: ;             bit 0 - Up
 8: ;                 1 - Down
 9: ;                 2 - Left
10: ;                 3 - Right
11: ;                 4 - Button A
12: ;                 5 -    "   B
13: ;                 6 -    "   C
14: ;                 7 -    "   D
15: ; break : AF
16: ; ==============================
17: KEY_READ:
18:         EXX
19:
20:         CALL    _GETKY          ; Key read
21:         OR      A
22:         JP      Z,JK_L2
23:
24:         LD      HL,(JKLIST_ADR)
25:         LD      B,12
26: JK_L0:
27:         CP      (HL)
28:         JR      Z,JK_L1         ; Key found !
29:         INC     HL
30:         DJNZ    JK_L0
31:
32:         XOR     A               ; Don't press joy key
33:
34:         EXX
35:         RET
36: JK_L1:
37:         LD      DE,(JKLIST_ADR)
38:         SBC     HL,DE
39:         LD      DE,KBIT_TBL     ; Key bit table address
40:         ADD     HL,DE
41:         LD      A,(HL)          ; Joy key data get
42: JK_L2:
43:         EXX
44:         RET
45:
46: ; ==============================
47: ;         KEY EDGE CHECK
48: ; ------------------------------
49: ; in    : A = New joykey data
50: ;         B = Old joykey data
51: ; out   : A = Key edge data
52: ; break : AF
53: ; ==============================
54: KEY_EDGE:
55:         PUSH    BC
56:
57:         LD      C,A
58:         LD      A,B
59:         XOR     C
60:         AND     C
61:
62:         POP     BC
63:         RET
64:
```

リスト6

```
 1: ; LIST 6
 2: ; ==============================
 3: ;         CHR GENERATE
 4: ; ------------------------------
 5: ; in    : non
 6: ; out   : non
 7: ; break : AF
 8: ; ==============================
 9: CHR_GEN:
10:         EXX
11:
12:         LD      HL,(SYS_COUNT)
13:         INC     HL              ; System count up
14:         LD      (SYS_COUNT),HL
15:
16:         LD      HL,(GENE_POS)
17: CG_L0:
18:         LD      DE,(SYS_COUNT)
19:
20:         LD      C,(HL)
21:         INC     HL
22:         LD      B,(HL)
23:         INC     HL
24:         EX      DE,HL
25:         OR      A               ; Carry clear
26:         SBC     HL,BC
27:         JR      NC,CG_L1
28:
29:         EXX
30:         RET
31: CG_L1:
32:         EX      DE,HL
33:         LD      A,(HL)          ; CHR number get
34:         INC     HL
35:         OR      A
36:         JP      M,CG_L3         ; System command
37:         LD      B,(HL)          ; Search count get
38:         INC     HL
39:         LD      E,(HL)          ; Offset address get
40:         INC     HL
41:         LD      D,(HL)          ;       "
42:         INC     HL
43:         PUSH    HL
44:         CALL    OBJ_SET         ; Work get
45:         JR      C,CG_L2
46:         PUSH    HL
47:         POP     IX
48:         POP     HL
49:         LD      B,(HL)          ; X position get
50:         INC     HL
51:         LD      C,(HL)          ; Y position get
52:         INC     HL
53:         LD      (IX+XPS),B
54:         LD      (IX+XPS+1),0
55:         LD      (IX+YPS),C
56:         LD      (IX+YPS+1),0
57:
58:         LD      (GENE_POS),HL   ; Generate position renewal
59:
60:         JP      CG_L0
61: CG_L2:
62:         POP     HL
63:         INC     HL
64:         INC     HL
65:         LD      (GENE_POS),HL   ; Generate position renewal
66:
67:         JP      CG_L0
68: CG_L3:
69:         LD      HL,1
70:         LD      (SYS_COUNT),HL  ; System count clear
71:         LD      HL,(GEN_TBL_ADR)
72:         LD      (GENE_POS),HL   ; Generate table position
73:
74:         JP      CG_L0
75:
```

リスト7

```
 1: ; LIST 7
 2: ; ==============================
 3: ;         CHR CONTROL
 4: ; ------------------------------
 5: ; in    : non
 6: ; out   : non
 7: ; break : ALL
 8: ; ==============================
 9: CHR_CTRL:
10:         CALL    HIT_INIT
11:
12:         LD      IX,CHR_WORK
13:         LD      B,64
14: CC_L0:
15:         LD      A,(IX)          ; Work use check
16:         OR      A
17:         JP      Z,CC_L1
18:
19:        ;INC     (IX+CNT)        ; Counter up
20:         LD      A,(IX+CNT)
21:         INC     A
22:         LD      (IX+CNT),A
23:
24:         LD      D,0
25:         LD      A,(IX+COD)      ; Get chr code
26:         ADD     A,A
27:         RL      D
28:         LD      E,A
29:         LD      HL,(CHR_TBL_ADR) ; User chr table address
30:         ADD     HL,DE
31:         LD      E,(HL)
32:         INC     HL
33:         LD      D,(HL)
34:         EX      DE,HL
35:
36:         PUSH    BC
37:         PUSH    IX
38:
39:         CALL    CC_L2
40:
```

▶「LOGIN」からもX68000が消えた……(編注:正確にはX68新聞)。私のX68000熱は冷めないが，世にこの子の価値を知ってもらえないのは悲しいナ。　藤沢　実(20)東京都

```
41:          POP     IX
42:          POP     BC
43:
44:          CALL    HIT_CHECK
45: CC_L1:
46:          LD      DE,32
47:          ADD     IX,DE
48:
49:          DJNZ    CC_L0
50:
51:          RET
52: CC_L2:
53:          JP      (HL)                    ; Target program call
54:
```

リスト8

```
  1: ; ==============================
  2: ;      OBJECT SCREEN CLEAR
  3: ; ------------------------------
  4: ; in    : non
  5: ; out   : non
  6: ; break : AF
  7: ; ==============================
  8: OBJ_CLR:
  9:         EXX
 10:
 11:         LD      HL,VRAM0
 12:         LD      A,' '
 13:         LD      B,0
 14: OR_L0:
 15:         LD      (HL),A
 16:         INC     HL
 17:         LD      (HL),A
 18:         INC     HL
 19:         LD      (HL),A
 20:         INC     HL
 21:         LD      (HL),A
 22:         INC     HL
 23:
 24:         LD      (HL),A
 25:         INC     HL
 26:         LD      (HL),A
 27:         INC     HL
 28:         LD      (HL),A
 29:         INC     HL
 30:         LD      (HL),A
 31:         INC     HL
 32:
 33:         DJNZ    OR_L0
 34:
 35:         EXX
 36:         RET
 37:
 38: ; ==============================
 39: ;      OBJECT DISPLAY
 40: ; ------------------------------
 41: ; in    : non
 42: ; out   : non
 43: ; break : AF
 44: ; ==============================
 45: OBJ_DISP:
 46:         PUSH    IX
 47:         EXX
 48:
 49:         LD      IX,CHR_WORK
 50:         LD      B,64
 51: OD_L0:
 52:         LD      A,(IX)          ; Work use check
 53:         OR      A
 54:         JP      Z,OD_L2         ; 0 = no use
 55:         JP      M,OD_L2         ; 0 > no disp
 56:
 57:         LD      HL,VRAM0        ; Player screen
 58:
 59:         LD      A,(IX+ATR)
 60:         OR      A
 61:         JP      P,OD_L1
 62:
 63:         LD      HL,VRAM1        ; Enemy screen
 64: OD_L1:
 65:         CALL    VRAM_ADR        ; VRAM address calc
 66:
 67:         LD      A,(IX)
 68:         CP      4
 69:         JP      Z,CHRN_PUT
 70:         CP      3
 71:         JP      Z,CHR3_PUT
 72:         CP      2
 73:         JP      Z,CHR2_PUT
 74:         JP      CHR1_PUT
 75: OD_L2:
 76:         LD      DE,32
 77:         ADD     IX,DE                   ; Next work address
 78:
 79:         DJNZ    OD_L0
 80:
 81:         EXX
 82:         POP     IX
 83:         RET
 84:
 85: ; ==============================
 86: ;      1x1 CHR PUT SUB
 87: ; ==============================
 88: CHR1_PUT:
 89:         LD      E,(IX+PAT)
 90:         LD      (HL),E
 91:         JP      OD_L2
 92:
 93: ; ==============================
 94: ;      2x2 CHR PUT SUB
 95: ; ==============================
 96: CHR2_PUT:
 97:         PUSH    BC
 98:
 99:         LD      E,(IX+PAT)
100:         LD      D,(IX+PAT+1)
101:
102:         EX      DE,HL
103:         LD      BC,32
104:         LDI
105:         LDI
106:         EX      DE,HL
107:         ADD     HL,BC
108:         EX      DE,HL
109:         LDI
110:         LDI
111:         POP     BC
112:
113:         JP      OD_L2
114:
115: ; ==============================
116: ;      3x3 CHR PUT SUB
117: ; ==============================
118: CHR3_PUT:
119:         PUSH    BC
120:
121:         LD      E,(IX+PAT)
122:         LD      D,(IX+PAT+1)
123:
124:         EX      DE,HL
125:         LD      BC,32
126:         LDI
127:         LDI
128:         LDI
129:         EX      DE,HL
130:         ADD     HL,BC
131:         EX      DE,HL
132:         LD      BC,32
133:         LDI
134:         LDI
135:         LDI
136:         EX      DE,HL
137:         ADD     HL,BC
138:         EX      DE,HL
139:         LDI
140:         LDI
141:         LDI
142:
143:         POP     BC
144:
145:         JP      OD_L2
146:
147: ; ==============================
148: ;      nxn CHR PUT SUB
149: ; ==============================
150: CHRN_PUT:
151:         PUSH    BC
152:
153:         LD      E,(IX+PAT)
154:         LD      D,(IX+PAT+1)
155:
156:         EX      DE,HL
157:
158:         LD      C,(HL)          ; X size get
159:         INC     HL
160:         LD      B,(HL)          ; Y size get
161:         INC     HL
162:
163:         LD      A,32
164:         SUB     C
165: CNP_L0:
166:         PUSH    BC
167:
168:         LD      B,0
169:         LDIR
170:         LD      C,A
171:         EX      DE,HL
172:         ADD     HL,BC
173:         EX      DE,HL
174:
175:         POP     BC
176:         DJNZ    CNP_L0
177:
178:         POP     BC
179:
180:         JP      OD_L2
181:
```

▶システム手帳を使い始めました。飲み屋やBARをデータベース化してシステム手帳サイズにプリントアウトしたいのですが，どのソフトがいいのやら……。「Multiword」と「Easydraw」ですかね？　あれ，「Easydraw」ってSX-WINDOWが必要でしたっけ？
内山　健(24)東京都

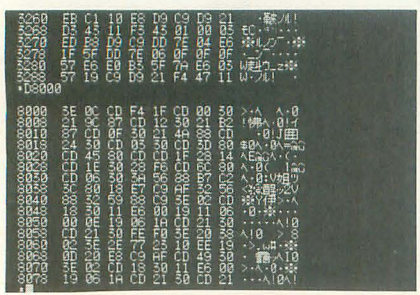
THE SENTINEL

全機種共通

S-OS"SWORD"要

怪しいZ80の使い方（未定義命令編）

Chikushi Takahiro
筑紫 高宏

今回は，予告どおりZ80の未定義命令のすべてを紹介します。また，Z80ハンドディスアセンブル表も用意しましたので，ダンプリストをがんがん読みこなしてください。

本題に入る前に，いつも私が利用しているZ80を扱うのならばぜひとも購入してほしい本を紹介します。それは，「Z80ファミリ・ハンドブック」（額田忠之著，CQ出版，定価1,900円，これは初版当時の値段）です。Z80の詳しい解説はもちろん，Z80周辺についても解説があり，価格もお手ごろです。Z80プログラマは，全員買ってください。Z80については，この本とZ80の裏マニュアルといえる，今回の「怪しいZ80の使い方(未定義命令編)」を押さえればバッチリです。

しかし，SIOを使用する場合，ある順番にレジスタを初期化しないといけないんですが，この本ではこの手のことを記述されてないようなことがチラホラあり，地獄を見ました。やはり，とことん詳しく書いてもらいたいものです。資料本全体がわかりやすいとか，正確とか以前の問題，「詳しくない」で苦労しています。

それと，サンプルプログラムについて。いままで「解説書」の理解を助けるオマケと思っていましたが，実際には本の内容のほとんどがサンプルプログラムで，肝心な「解説」の部分がオマケ程度という本があったりします。皆さんは，このような本を購入しないようにしましょう。

ここで，よけいなお世話かもしれませんが，「Z80ファミリ・ハンドブック」の，Z80解説のタイムステートの部分の間違いを書いておきます。訂正版が出ていないかもしれないので，参考にしてください。

123ページ「RET cc」のマシンサイクル数とタイムステート数は，条件成立時のもので，条件不成立時は，「M＝1，T＝5」です。ちなみに，これは珍しく1バイト命令で5クロック待ちができるので，時間待ちに重宝します。

144ページの「XOR r」は，その上にある「XOR(IY＋d)」と混同してしまっているようで，実際は「M＝1，T＝4」です。

Z80のバグ

Z80のバグですが，未定義命令では，いまのところ報告されていません。正式命令では，「LD A，I」と「LD A,R」のP/Vフラグの変化（これは，現在の割り込みの状態を示すものです）にバグがあり，正しい状態を示さない場合があります（X1平和研究所からの報告）。シャープ製のZ80，MSXエンジンで確認できました。

ちなみに，MSXではインタースロットコールで，この命令を使い「呼び出したルーチンのコール後の割り込みの状態」を保存しようとしていますが(BIOSのプログラムは，このように設計されています。結局，無駄な時間とスタックを消費しているわけです)，このバグにより割り込み禁止か許可かが確定しない状態になっています。

よって，本来ならキーを受けつけなくなって，プッツンするはずですが（インタースロットコール後，通常，割り込み許可を実行しないため），VDPのアクセスなどで，ひんぱんに「DI」や「EI」を実行しているので，運よくプッツンしないようです。それにしても，このインタースロットコール，通常，1500～2000クロックかかるんですよね，インタースロットコールの処理部分だけで。どうにかならなかったのかな。

またまた，よけいなお世話ですが「MSX2テクニカル・ハンドブック」の339ページの「CALSLT使用時の注意」には，5カ所の間違いがあります。詳しい解説はしませんが，対処法は，インタースロットコール後（スロット関連全部），必ず「EI」で割り込み許可してください（アスキーの本のサンプルプログラムでは，たいてい「EI」していないようです）。

さらに，MSXの仕様を決定した人と，解説書を書いた人との連絡がうまくいっていない部分があるようで，実際のルーチンと，破壊レジスタ，ワークアドレス，ワークの値，ルーチンの仕様が違っていることがあるようです。ちなみに，「MSXturboRテクニカル・ハンドブック」38ページPCMのサンプリングレート「15.7500kHz」ですが，循環小数の水晶発信器を分周するので，このような数値にはならないはずです（割り切れないから)。X68000は，16MHzを分周するので，きっちり，15.625Hzになりますが。

Z80未定義命令

さて，未定義命令ですが，「I/O」1986年4月号のやはぎたかし氏の「超高速逆アセンブラ」にほとんど載ってました。それを「MSXマガジン」1988年4月号の「マシン語

プログラミング入門」に掲載されていた，CPU命令のチェックプログラムをX1に移植して，命令の動作とフラグ変化をチェックしました。さらに，プログラムを作り，実行時間も調べて，新たに発見した未定義命令も書きます。たぶん，これで全部だと思います。

正式命令でも，IXやIYのディスプレイスメントつき「LD A, (IX＋n)」などは，遅すぎてほとんど使用されませんが，未定義命令のうちIXやIYを8ビットのIXH, IXL, IYH, IYLに分解して使用できるもの，および，ディスプレイスメントのないものは有用なので，よく使用されます。

Z80は，だいたいM1サイクルでは，4クロック，メモリアクセスで3クロック，相対アドレス計算で5クロックかかります。R800では，メモリアクセス，相対アドレス計算は1クロックかかるようです。

未定義命令が存在するのは，エスケープシーケンスのように，CB_H, ED_H, DD_H, FD_Hに続くものです。DD_H, FD_Hはインデックスレジスタ関係，CB_Hはビット操作関係，ED_Hはその他拡張命令，という具合に使用されています。

●IN F, (C)

I/Oポートより，データを読み込んで，値をレジスタに格納せずに，フラグのみ変化させる命令です。マシンコードは「ED 70」です。フラグ変化とマシンサイクル数，タイムステート数は，「IN r, (C)」命令と同じです。R800でも，正式にサポートされています。

●OUT(C), 0

I/Oポートに，常に「0」を出力します。マシンコードは「ED 71」です。フラグ変化などは，「OUT (C), r」と同じです。R800での動作は不明です。

●SLI

レジスタ，または，メモリを左にローテイトして，b0に必ず「1」が入る（CY←b7……b0←1）という命令で，実行前のb7がCYに入る以外はSRL命令と，フラグ変化などが同じです。R800でサポートされているかは不明です。

マシンコードは，「SLI r」（rはB, C, D, E, H, L, A）は「CB　n」（nは30～35, 37）です。

「SLI (HL)」は「CB 36」。

「SLI (IX＋d)」は「DD CB d 36」，「SLI (IY＋d)」は「FD CB d 36」です。

●IXH, IXL, IYH, IYL関係

インデックスレジスタは，上位バイトと下位バイトを分離し，8ビットのレジスタ(IXの上位がIXHなど）として使用できます。HレジスタやLレジスタを，8ビットレジスタとして使用する命令の前に，DD_H (IX)やFD_H (IY)を付加すると，HやLの部分が，IXHかIYH, IXLかIYLにそっくり置き換わるわけです。ディスプイレスメントはつきません。フラグ変化は，もとになる命令と同じで，マシンサイクル数は1つ増え，タイムステート数は4増えます。この命令は，結構使えるもので，8ビットレジスタが足りなくなった場合や，外側のループカウンタなどに使います。「DEC IXH」などとするわけです。

インデックスレジスタを8ビットレジスタとして使用できる命令は，「INC r」「DEC r」「LD r, n」「LD r, r'」「ADD A, r」「ADC A, r」「SUB r」「SBC A, r」「AND r」「XOR r」「OR r」「CP r」です。

●複合命令

インデックスレジスタを使い，2つの動作を複合した未定義命令があります。R800でサポートされているかは不明です。

複合命令は，ローテイトやビット操作命令のあと，結果を8ビットレジスタ（B, C, D, E, H, L, A）へロードするものです。

対応する命令は，「RLC r」「RRC r」「RL r」「RR r」「SLA r」「SRA r」「SLI r」「SRL r「RES b, r」「SET b, r」です。これらの命令の前にDD_HやFD_Hをつけると，rの部分が「(IX＋d)」や「(IY＋d)」に置き換わり，その命令を実行後，「r」へ結果をロードします。なお，マシンコードの3バイト目にディスプレイスメントが割り込みます。

例を挙げると，「RLC B」(マシンコード「CB 00」，M＝2，T＝8（4，4))の前にDD_Hをつけると，「LD B, RLC(IX＋d)」(マシンコード「DD CB d 00」，M＝6，T＝23（4，4，3，5，4，3))となり，「RL C(IX＋d)」を実行したあと，「LD B, (IX＋d)」を行ったと同じ結果になります。

ED_Hに続く，40_H～$7F_H$の命令の空きは，既存の命令と同じ動作をするものがほとんどです。いわゆる「デコードされてない」状態だと思われます。

・ED n（nは，4C, 54, 5C, 64, 6C, 74, 7C）……「NEG」と同じ動作

・ED n（nは，55, 65, 75）……「RETN」と同じ動作

・ED n（nは，5D, 6D, 7D）……「RETI」と同じ動作

・ED n（nは，4E, 66, 6E）……「IM 0」と同じ動作

・ED 76……「IM 1」と同じ動作

・ED 7E……「IM 2」と同じ動作

・ED 77, ED 7F……未使用

「SLI」命令で，CB_H平面は埋め尽くされました。

さらにED_H平面では，00_H～$3F_H$, 77_H, $7F$, 80_H～$9F_H$, $A4_H$～$A7_H$, AC_H～AF_H, $B4_H$～$B7_H$, BC_H～BF_H, $C0_H$～FF_Hが未使用のようです。

R800では，掛け算命令の「mulub」と「muluw」が追加されました。「ED C1」など，ED_H平面の，$C0_H$～FF_Hのエリアが使用されています。

おまけのテクニック

祝一平氏の「試験に出るX1」でも紹介されていましたが，I/Oのカウンタつき命令では，Bレジスタは先にデクリメントされる，ということを覚えておきましょう。

さて，「OUT (C), A」などの命令は，アドレスバスにBCの値が出力されています。X1では，これを利用して64KバイトのI/Oエリアをもっているのです。さらにいうと「OUT (n), A」などの命令では，Aレジスタの値がアドレスバスの上位に出力されるので，X1では通常，

```
	LD	BC, 1A01H
	IN	A, (C)
```

と書かれるところを，

```
	LD	A, 1AH
	IN	A, (01H)
```

と書いても，だいたい同じ動作をするわけです。このようにすることで，BCレジスタを破壊せず，22.2%速く処理できます。サイズも1バイト小さいので，覚えておくといいかもしれません。

最後にZ80ハンドディスアセンブル表をまとめてみましたので，ダンプリストを読むときに利用してください。

Z80ハンドディスアセンブル表

これは，Z80マシンコードと，ニーモニックの対応表です。

未定義命令を含み，小文字で表記してあるのが，R800のコードです。

「*」印がついているものが，HLレジスタをインデックスレジスタに切り替えて使用できる命令です（動作は，若干異なる場合があります）。

「+」印がついているものが，操作対象のレジスタが，「(IX+n)」または「(IY+n)」に置き換わり，結果をもとの操作対象であったはずのレジスタへロードします。

複合命令は，「DD_H，CB_H，ディスプレイスメント，〜」(IXを使用する場合）というコード列で使用します。

ディスプレイスメントは，必ず，コード列の3バイト目に位置します。

「#」印がついているものは，8080Aにはなく，Z80で拡張された，M1サイクルが複数ある命令です。「#」の次に示してあるマシンコードから命令が始まります。Z80は，8080Aから，フラグが一部拡張されたらしいので，一部，フラグ変化に互換性がないかもしれません。また，割り込み状態はZ80のバグのため，正しく参照できないことがあります。

マシンコードの見方は，

00	10	20	30
40	50	60	70
80	90	A0	B0
C0	D0	E0	F0

上位4ビットが以上のように並んでいて，下位4ビットが表に示される[OFFSET nn]の値となります。つまり，「LD B,C」のマシンコードは，

$40_H+01_H=41_H$

という具合になります。

〈基本的な命令〉

[OFFSET 00_H]

NOP	DJNZ n	JR NZ,n	JR NC,n
LD B,B	LD D,B	*LD H,B	*LD (HL), B
ADD	SUB B	AND B	OR B
RET NZ	RET NC	RET PO	RET P

[OFFSET 01_H]

LD BC,nn	LD DE,nn	*LD HL,nn	LD SP,nn
LD B,C	LD D,C	*LD H,C	*LD (HL), C
ADD A,C	SUB C	AND C	OR C
POP BC	POP DE	*POP HL	POP AF

[OFFSET 02_H]

LD (BC), A	LD (DE), A	*LD (nn), HL	LD (nn), A
LD B,D	LD D,D	*LD H,D	*LD (HL), D
ADD A,D	SUB D	AND D	OR D
JP NZ,nn	JP NC,nn	JP PO nn	JP P,nn

[OFFSET 03_H]

INC BC	INC DE	*INC HL	INC SP
LD B,E	LD D,E	*LD H,E	*LD (HL), E
ADD A,E	SUB E	AND E	OR E
JP nn	OUT (n), A	*EX (SP) ,HL	DI

[OFFSET 04_H]

INC B	INC D	*INC H	*INC (HL)
*LD B,H	*LD D,H	*LD H,H	*LD (HL), H
*ADD A,H	*SUB H	*AND H	*OR H
CALL NZ,nn	CALL NC,nn	CALL PO nn	CALL nn

[OFFSET 05_H]

DEC B	DEC D	*DEC H	*DEC (HL)
*LD B,L	*LD D,L	*LD H,L	*LD (HL), L
*ADD A,L	*SUB L	*AND L	*OR L
PUSH BC	PUSH DE	*PUSH HL	PUSH AF

[OFFSET 06_H]

LD B,n	LD D,n	*LD H,n	*LD (HL), n
*LD B,(HL)	*LD D,(HL)	*LD H, (HL)	HALT
*ADD A,(HL)	*SUB (HL)	*AND (HL)	*OR (HL)
ADD A,n	SUB n	AND n	OR n

[OFFSET 07_H]

RLCA	RLA	DAA	SCF
LD B,A	LD D,A	*LD H,A	*LD (HL), A
ADD A,A	SUB A	AND A	OR A
RST 00H	RST 10H	RST 20H	RST 30H

[OFFSET 08_H]

EX AF,AF'	JR n	JR Z,n	JR C,n
LD C,B	LD E,B	*LD L,B	LD A,B
ADC A,B	SBC A,B	XOR B	CP B
RET Z	RET C	RET PE	RET M

[OFFSET 09_H]

*ADD HL,BC	*ADD HL,DE	*ADD HL,HL	*ADD HL,SP
LD C,C	LD E,C	*LD L,C	LD A,C
ADC A,C	SBC A,C	XOR C	CP C
RET	EXX	*JP (HL)	*LD SP,HL

[OFFSET $0A_H$]

LD A, (BC)	LD A,(DE)	*LD HL,(nn)	LD A,(nn)
LD C,D	LD E,D	*LD L,D	LD A,D
ADC A,D	SBC A,D	XOR D	CP D
JP Z,nn	JP C,nn	JP PE,nn	JP M,nn

[OFFSET $0B_H$]

DEC BC	DEC DE	*DEC HL	DEC SP
LD C,E	LD E,E	*LD L,E	LD A,E
ADC A,E	SBC A,E	XOR E	CP E
#CBH	IN A,(n)	EX DE,HL	EI

[OFFSET $0C_H$]

INC C	INC E	*INC L	INC A
*LD C,H	*LD E,H	*LD L,H	*LD A,H
*ADC A,H	*SBC A,H	*XOR H	*CP H
CALL Z,nn	CALL C,nn	CALL PE,nn	CALL M,nn

[OFFSET $0D_H$]

DEC C	DEC E	*DEC L	DEC A
*LD C,L	*LD E,L	*LD L,L	*LD A,L
*ADC A,L	*SBC A,L	*XOR L	*CP L
CALL nn	#DDH	#EDH	#FDH

[OFFSET $0E_H$]

LD C,n	LD E,n	*LD L,n	LD A,n
*LD C,(HL)	*LD E,(HL)	*LD L,(HL)	*LD A,(HL)
*ADC A,(HL)	*SBC A,(HL)	*XOR (HL)	*CP (HL)
ADC A,n	SBC A,n	XOR n	CP n

[OFFSET $0F_H$]

RRCA	RRA	CPL	CCF
LD C,A	LD E,A	*LD L,A	LD A,A
ADC A,A	SBC A,A	XOR A	CP A
RST 08H	RST 18H	RST 28H	RST 38H

〈ED_Hに続く命令〉

[OFFSET 00_H]

x（空き）	x	x	x
IN B, (C)	IN D, (C)	IN H, (C)	IN F, (C)
x	x	LDI	LDIR
x	x	x	x

[OFFSET 01_H]

x	x	x	x
OUT (C),B	OUT (C),D	OUT (C),H	OUT (C),0
x	x	CPI	CPIR
mulub a,b	mulub a,d	x	x

[OFFSET 02_H]

x	x	x	x
SBC HL,BC	SBC HL,DE	SBC HL,HL	SBC HL,SP
x	x	INI	INIR
x	x	x	x

[OFFSET 03_H]

x	x	x	x
LD (nn),BC	LD (nn),DE	LD (nn),HL	LD (nn),SP
x	x	OUTI	OTIR
muluw hl,bc	x	x	muluw hl,sp

[OFFSET 04_H]

x	x	x	x
NEG	NEG	NEG	NEG
x	x	x	x
x	x	x	x

[OFFSET 05_H]

x	x	x	x
RETN	RETN	RETN	RETN
x	x	x	x
x	x	x	x

[OFFSET 06_H]

x	x	x	x
IM0	IM1	IM0	IM1
x	x	x	x
x	x	x	x

[OFFSET 07_H]

x	x	x	x
LD I,A	LD A,I	RRD	-
x	x	x	x
x	x	x	x

[OFFSET 08_H]

x	x	x	x
IN C,(C)	IN E,(C)	IN L,(C)	IN A,(C)
x	x	LDD	LDDR
x	x	x	x

[OFFSET 09_H]

x	x	x	x
OUT (C),C	OUT (C),E	OUT (C),L	OUT (C),A
x	x	CPD	CPDR
mulub a,c	mulub a,e	x	x

[OFFSET $0A_H$]

x	x	x	x
ADC HL,BC	ADC HL,DE	ADC HL,HL	ADC HL,SP
x	x	IND	INDR
x	x	x	x

[OFFSET $0B_H$]

x	x	x	x
LD BC,(nn)	LD DE,(nn)	LD HL,(nn)	LD SP,(nn)
x	x	OUTD	OTDR
x	x	x	x

[OFFSET $0C_H$]

x	x	x	x
NEG	NEG	NEG	NEG

x	x	x	x
x	x	x	x

[OFFSET $0D_H$]

x	x	x	x
RETI	RETI	RETI	RETI
x	x	x	x
x	x	x	x

[OFFSET $0E_H$]

x	x	x	x
IM0	IM2	IM0	IM2
x	x	x	x
x	x	x	x

[OFFSET $0F_H$]

x	x	x	x
LD R,A	LD A,R	RLD	-
x	x	x	x
x	x	x	x

〈CB_Hに続く命令〉

[OFFSET 00_H]

+RLC B	+RL B	+SLA B	+SLI B
BIT 0,B	BIT 2,B	BIT 4,B	BIT 6,B
+RES 0,B	+RES 2,B	+RES 4,B	+RES 6,B
+SET 0,B	+SET 2,B	+SET 4,B	+SET 6,B

[OFFSET 01_H]

+RLC C	+RL C	+SLA C	+SLI C
BIT 0,C	BIT 2,C	BIT 4,C	BIT 6,C
+RES 0,C	+RES 2,C	+RES 4,C	+RES 6,C
+SET 0,C	+SET 2,C	+SET 4,C	+SET 6,C

[OFFSET 02_H]

+RLC D	+RL D	+SLA D	+SLI D
BIT 0,D	BIT 2,D	BIT 4,D	BIT 6,D
+RES 0,D	+RES 2,D	+RES 4,D	+RES 6,D
+SET 0,D	+SET 2,D	+SET 4,D	+SET 6,D

[OFFSET 03_H]

+RLC E	+RL E	+SLA E	+SLI E
BIT 0,E	BIT 2,E	BIT 4,E	BIT 6,E
+RES 0,E	+RES 2,E	+RES 4,E	+RES 6,E
+SET 0,E	+SET 2,E	+SET 4,E	+SET 6,E

[OFFSET 04_H]

+RLC H	+RL H	+SLA H	+SLI H
BIT 0,H	BIT 2,H	BIT 4,H	BIT 6,H
+RES 0,H	+RES 2,H	+RES 4,H	+RES 6,H
+SET 0,H	+SET 2,H	+SET 4,H	+SET 6,H

[OFFSET 05_H]

+RLC L	+RL L	+SLA L	+SLI L
BIT 0,L	BIT 2,L	BIT 4,L	BIT 6,L
+RES 0,L	+RES 2,L	+RES 4,L	+RES 6,L
+SET 0,L	+SET 2,L	+SET 4,L	+SET 6,L

[OFFSET 06_H]

*RLC (HL)	*RL (HL)	*SLA (HL)	*SLI (HL)
*BIT 0,(HL)	*BIT 2,(HL)	*BIT 4,(HL)	*BIT 6,(HL)
*RES 0,(HL)	*RES 2,(HL)	*RES 4,(HL)	*RES 6,(HL)
*SET 0,(HL)	*SET 2,(HL)	*SET 4,(HL)	*SET 6,(HL)

[OFFSET 07_H]

+RLC A	+RL A	+SLA A	+SLI A
BIT 0,A	BIT 2,A	BIT 4,A	BIT 6,A

+RES 0,A	+RES 2,A	+RES 4,A	+RES 6,A
+SET 0,A	+SET 2,A	+SET 4,A	+SET 6,A
[OFFSET 08$_H$]			
+RRC B	+RR B	+SRA B	+SRL B
BIT 1,B	BIT 3,B	BIT 5,B	BIT 7,B
+RES 1,B	+RES 3,B	+RES 5,B	+RES 7,B
+SET 1,B	+SET 3,B	+SET 5,B	+SET 7,B
[OFFSET 09$_H$]			
+RRC C	+RR C	+SRA C	+SRL C
BIT 1,C	BIT 3,C	BIT 5,C	BIT 7,C
+RES 1,C	+RES 3,C	+RES 5,C	+RES 7,C
+SET 1,C	+SET 3,C	+SET 5,C	+SET 7,C
[OFFSET 0A$_H$]			
+RRC D	+RR D	+SRA D	+SRL D
BIT 1,D	BIT 3,D	BIT 5,D	BIT 7,D
+RES 1,D	+RES 3,D	+RES 5,D	+RES 7,D
+SET 1,D	+SET 3,D	+SET 5,D	+SET 7,D
[OFFSET 0B$_H$]			
+RRC E	+RR E	+SRA E	+SRL E
BIT 1,E	BIT 3,E	BIT 5,E	BIT 7,E
+RES 1,E	+RES 3,E	+RES 5,E	+RES 7,E

+SET 1,E	+SET 3,E	+SET 5,E	+SET 7,E
[OFFSET 0C$_H$]			
+RRC H	+RR H	+SRA H	+SRL H
BIT 1,H	BIT 3,H	BIT 5,H	BIT 7,H
+RES 1,H	+RES 3,H	+RES 5,H	+RES 7,H
+SET 1,H	+SET 3,H	+SET 5,H	+SET 7,H
[OFFSET 0D$_H$]			
+RRC L	+RR L	+SRA L	+SRL L
BIT 1,L	BIT 3,L	BIT 5,L	BIT 7,L
+RES 1,L	+RES 3,L	+RES 5,L	+RES 7,L
+SET 1,L	+SET 3,L	+SET 5,L	+SET 7,L
[OFFSET 0E$_H$]			
*RRC (HL)	*RR (HL)	*SRA (HL)	*SRL (HL)
*BIT 1, (HL)	*BIT 3, (HL)	*BIT 5, (HL)	*BIT 7, (HL)
*RES 1, (HL)	*RES 3, (HL)	*RES 5, (HL)	*RES 7, (HL)
*SET 1, (HL)	*SET 3, (HL)	*SET 5, (HL)	*SET 7, (HL)
[OFFSET 0F$_H$]			
+RRC A	+RR A	+SRA A	+SRL A
BIT 1,A	BIT 3,A	BIT 5,A	BIT 7,A
+RES 1,A	+RES 3,A	+RES 5,A	+RES 7,A
+SET 1,A	+SET 3,A	+SET 5,A	+SET 7,A

命令のクロックは? (PART2)

さっそく先月号の続きを紹介します。

●デクリメントジャンプノンゼロ

DJNZ　e

Bレジスタ=0のとき8クロック, Bレジスタ≠0のとき13クロック

●サブルーチンコール

・無条件サブルーチンコール

CALL　Im

以上, 17クロック

・条件付サブルーチンコール

CALL　NZ,Im

(設定条件=Z,NC,C,PO,PE,P,M)

条件成立時17クロック, 不成立時10クロック

●リターン命令

・無条件リターン

RET

以上, 10クロック

・条件付リターン

RET　NZ

(設定条件=Z,NC,C,PO,PE,P,M)

条件成立時11クロック, 不成立時5クロック

●スタック操作命令

・PUSH命令

PUSH　AF (BC,DE,HL)

以上, 11クロック

PUSH　IX

PUSH　IY

以上, 15クロック

・POP命令

POP　AF (BC,DE,HL)

以上, 10クロック

POP　IX

POP　IY

以上, 14クロック

●算術演算命令

・8ビット加算命令 (ADC命令も同様)

ADD　A,A~L

以上, 4クロック

ADD　A,n

ADD　A,(HL)

以上, 7クロック

ADD　A,(IX+d)

ADD　A,(IY+d)

以上, 19クロック

・8ビット減算命令 (SBC命令も同様)

SUB　A,A~L

以上, 4クロック

SUB　A,n

SUB　A,(HL)

以上, 7クロック

SUB　A,(IX+d)

SUB　A,(IY+d)

以上, 19クロック

・比較命令

CP　A~L

以上, 4クロック

CP　n

CP　(HL)

以上, 7クロック

CP　(IX+d)

CP　(IY+d)

以上, 19クロック

・8ビットインクリメント命令 (8ビットデクリメント命令も同様)

INC　A~L

以上, 4クロック

INC　(HL)

以上, 11クロック

INC　(IX+d)

INC　(IY+d)

以上, 23クロック

・Aレジスタ10進補正

DAA

以上, 4クロック

・ニゲイト (Aレジスタの2の補数をとる)

NEG

以上, 4クロック

・16ビット加算命令

ADD　HL,BC (DE,HL,SP)

以上, 11クロック

ADD　IX,BC (DE,IX,SP)

ADD　IY,BC (DE,IY,SP)

以上, 15クロック

・16ビットキャリつき加算命令 (減算命令も同じ)

ADC　HL,BC (DE,HL,SP)

以上, 15クロック

・16ビットインクリメント (16ビットデクリメント命令も同様)

INC　BC (DE,HL,SP)

以上, 6クロック

INC　IX

INC　IY

以上, 10クロック

●論理演算命令

・論理積(論理和OR命令, 排他的論理和XOR命令も同様)

AND　A~L

以上, 4クロック

AND　n

AND　(HL)

以上, 7クロック

AND　(IX+d)

AND　(IY+d)

以上, 19クロック

・Aレジスタビット反転

CPL

以上, 4クロック

愛読者プレゼント

1 レッスルエンジェルス3

5名

X68000用

5″2HD版　5,800円（税込）

8月号で紹介したカード型バトルゲームの3作目。今回はプレイヤーは女子プロレスの団体を経営し，レスラーを育てるのだ。好評につき，次回作の発売も決定。

▲TAKERU　☎052(824)2493

2 最新MIDI DATA制作術　基礎編・実践編

3名　各3,000円（税込）

9月号のペンギン情報コーナーで紹介した書籍です。SC-55とPC-9801用レコンポーザを使ってMIDIデータの制作を解説します。2冊セットで3名にプレゼント。

▲東亜音楽社　☎03(3260)6271

3 「満開の電子ちゃん」

5名

大人気の電子ちゃんが単行本になりました。オフセット印刷，表紙はカラーの豪華本。夏のコミケットで販売されたものです。

▲電子出版

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1994年10月18日の到着分までとします。当選者の発表は1994年12月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞には当選できない場合がありますので，ご了承ください。

4 「ツクモの日」記念テレホンカード

10名　非売品

ツクモ電機では，9月9日と19日を「ツクモの日」として全国のショップでキャンペーンを展開中。そのCFキャラクターの越智静香の記念テレカを10名に。どのデザインになるかは当たってからのお楽しみね。

▲ツクモ電機

5 新雑誌創刊記念テレホンカード

10名　非売品

9月8日に「Hello! PC」，10月8日に「GAMEBLAST」と相次いでソフトバンクより新雑誌創刊。Oh!X同様かわいがってね，ということで記念テレカをプレゼント。

▲ソフトバンク　☎03(5642)8100

8月号プレゼント当選者

1 スーパーリアル麻雀PⅣ（岡山県）真方隆志　2 雀神クエスト（北海道）森谷好雄（神奈川県）近藤健一（広島県）山崎隆司　3 麻雀航海記（宮城県）吉田淳一（茨城県）姉帯　寛（福岡県）佐村和亮　4 「ツインビーPARADISE」下敷き　A（宮城県）鈴木政宏（千葉県）伊藤正章　佐久間利浩　B（群馬県）久保田智久（東京都）田幡　厚（岡山県）杉浦竜夫　（敬称略）
以上の方々が当選しました。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。

玉石混交の山の中で輝く妙なソフト

新マシンがわが家に

迷いに迷いましたが，意を決して自宅用にMacintosh(以下マック)を買いました。LC575というモデルです。いままで家ではPowerBook170を使っていたのですが，家でもしっかりと仕事をしたり，グラフィックやゲームなども心おきなくできる頼りになるマックがほしくなってきたのです。

どの機種にするかということではかなり迷いました。そして，1）スピードがそこそこ出て，2）コストパフォーマンスがよく，3）比較的コンパクトにまとまっているという3つの条件を満たしているということでLC575に決まりました。

まず，スピードに関してですが，プロセッサは68LC040の33MHzですから，僕が大学で机の上に置いているQuadra700よりクロックの分だけ速いということになります。ただ残念ながら，FPU（浮動小数点演算ユニット）を装備していないということが，ときどき大きなハンディになるということがわかってきました。

カラーモニタ，CD-ROM，320Mバイトハードディスク，メモリ20Mバイト，キーボードなどを含んで買い値で30万円を切っているのですから，かなりコストパフォーマンスはよいと感じられるのですがどうなのでしょう？

アップル社のこれまでの価格設定に麻痺してしまったからそう思うということはないでしょうね。

次にコンパクトにまとまっているかということについてです。マックを使うための専用の部屋があるのならば話は別ですが，ごちゃごちゃとした部屋に，本体，ディスプレイ，CD-ROM，キーボード，マウス，プリンタ，モデム，ステレオスピーカ，マイク，購入予定のスキャナなどがごろごろするのですからたいへんです。さらにこの家には，まだわけのわからないこびととかが住んでいるのですから……。

デザインは「オールインワン」

LC575はディスプレイ一体型になっています。外観デザイン面についてはどこを評価していいかわからないというか，ろくでもないデザインに見えても，ひとつに寄せ集まっているとそれなりに親近感が湧くところが不思議です。ばらばらだと，まったくとりとめもないものが，オールインワンということでひとつになる(キーボードはケーブルでつながっているのですが)と全体としての存在を主張し始めるのです。

あるいは，万能なものに対するあこがれのようなものが根本のところに見えてくる気がします。スピーカーから音が聞こえ，モニタから光が出るのは当たり前ですけれども，ひとつの箱から，音が出たり，光が出たり，計算したり，CD-ROMをくわえこんだりすると嬉しいのです。

本体ひとつで，マックが走り，また同時にX68000も走れば嬉しいですよね。たとえ，中身を開ければマックとX68000が1台ずつ入っていて，しかも値段と体積が2倍近くても(そんなことはないか？)。

そうはいうものの，いくら見てもこのボディは不格好です。このデザインを考えた人が模型を作って，おエラいさんに見せる場面を想像して，「ほう，なかなかじゃのう！」とか「なるほど」などというレスポンスが出るシーンがどうしても浮かんできません。デザイン経費を落としたためというよりも，わざと不格好にしているのかと勘ぐりたくなってしまいます。

illustration：Haruhisa Yamada

チューンナップの2週間

宅急便屋さんがマックを運んできてからがたいへんです。最近は大学の授業の関係もあって，「マックずっぽし状態」になっているので，マックを自分に最適な環境にするために，ついリキが入ってしまいます。

大学の机の上のQuadra700とは画面の大きさが違う(大学のは大きい)し，そもそも用途が違うので，チューンナップの仕方もかなり違ってきます。システムのパラメータを調整したり，ネットワークでフリーソフト/PDSを取ってきたりして，少しずつ快適な環境を作り上げます。

家族(こびとも含む)で使うので，各ユーザーそれぞれの使い方を想定して設定します。こびとの場合は，画面上の「こびと」

と書かれたアイコン(SerenityというPDS)をダブルクリックすると，こびと用のプログラムだけがわかりやすく並んだデスクトップに変わります。そこに並んだプログラムの起動は簡単ですし，パスワードを打たないと，もとの画面(ファインダ)には戻らないようになっています。

フォルダのアイコンなども1つひとつ厳密に選定(あるいは作成)してあります。2週間ほどかけて，かなり満足のいく環境ができてきました。そして，いまこの深夜にも，楽しく原稿を書いています，小西師(芸術方面の私の師匠で簡単に手に入らない音楽CDを毎年40-50枚貸してくださいます。ちなみに師はAMIGAユーザーです)から借りた音楽CDを本体内蔵のステレオスピーカーから鳴らしながら。

ただ1点だけ，どうしてももの足らなさが残るのが，ネットワーク環境がないということです。大学のようにLANに接続されていないので，インターネットを利用できないということです(最近はお金次第で自宅にも自由にネットワーク環境を構築できるようになってきているようですが)。

モデムで何とかメールを読み書きしたり，ニュースをトロトロと読んだりはできますが，たとえば，この環境を作った際に利用したフリーソフト/PDSなどを検索したり，取り寄せたりすることが簡単にはできないからです。

たまげたCD-ROM本

『HyperLib No.2』(アスキー刊)というびっくりするような2,800円の雑誌(ムック)があります。何がすごいかというと，CD-ROMが2枚ついていて，それにアメリカのAMUGというユーザーグループの溜め込んだ3483本ものファイル(PDS)が入っているのです。3483本(フォントの1089本をひいても2394本)といったらすごいですよ。

PDSの種類によっては試して気に入り，定常的に使うようになったらお金を払わなくてはいけない場合がありますが，それを考えにいれなかったら，ソフト1本あたり1円ですよ，1円。大ざっぱな感覚ですが，このCD-ROMに含まれているソフトの価値は全部で1,000万円を軽～く越えるのは間違いないと思います。

この本を買ってしまったため，お盆の2晩徹夜して，ゲーム関係と音楽関係の計637本を全部試しました。すでに知っているものはむろん動かしませんし，ルールや使い方の複雑そうなものはすぐにやめましたが，とにかく圧倒的な分量でした。玉石混交でしたが，かなりのレベルを保っていることに驚きました(ソフトハウスが流すデモバージョンも結構ありましたが)。

計算機によるゲームというかアミューズメント関係には，大きな期待をもっています。いまでも，ゲームしかできないとか，ゲームばかりとかいうように，卑下されることもあるようですが，新しく出てきたメディアや文化のジャンルは，いつの時代でもまず卑下されてきたものです。

テレビや映画，ラジオもそうでした。マンガもそうでしょうが，パソコンゲームも文化というか芸術というか，本や映画などを乗り越えるようなメディアとして認知される日がすぐに来るでしょう。

という話はさておき，その試したソフトで特に印象に残るものをこれから紹介しましょう。単によくできて面白いというのは，ほうっておいてもどこかで紹介される(されている)でしょうから，ここではそうそう紹介する人もいないような，しかし，今後のゲームソフトの動向を占えるような，あるいは，大げさにいうならば人間とマシンとの今後の関係を考えさせてくれるようなソフトを紹介することにします。

●宗教関係

purity test

純潔度テストということで，興味本意でやり始めたら，すぐにこれは男のやるものではなくて，若い女性を想定しているということに気がつきました(当たり前か？)。質問(もちろん英語)があまりに露骨ですごいのでやめられませんでした。

一般の教会でもこのような質問が用意されていると書いてあります。一応，まじめにやったところ，最後に判定が出て，あんたは並のアメリカ人だ，どんな仕事にも向いていると出ました。うーむ，すごいなアメリカ人は，だって対象は10代の後半の女性といったところでしょ。

試しに，完全な不純を装って答えていくと，あんたは望みがないほど堕落している，好きにしろ，ストリッパーか，低俗週刊誌の編集に向いていると出ました。懲りずにというか，余計面白がって，完全な純潔を装って返事をしていったら，あんたは完全な大嘘つきか，救世主になるか，中古車のセールスに向いていると出ました。すごいですね。

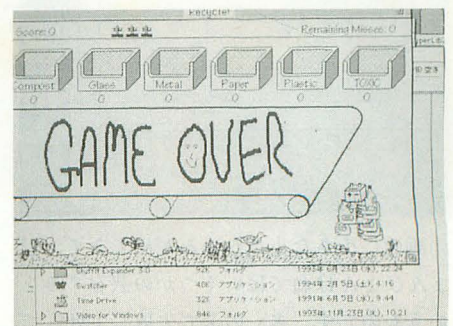
●ジョーク

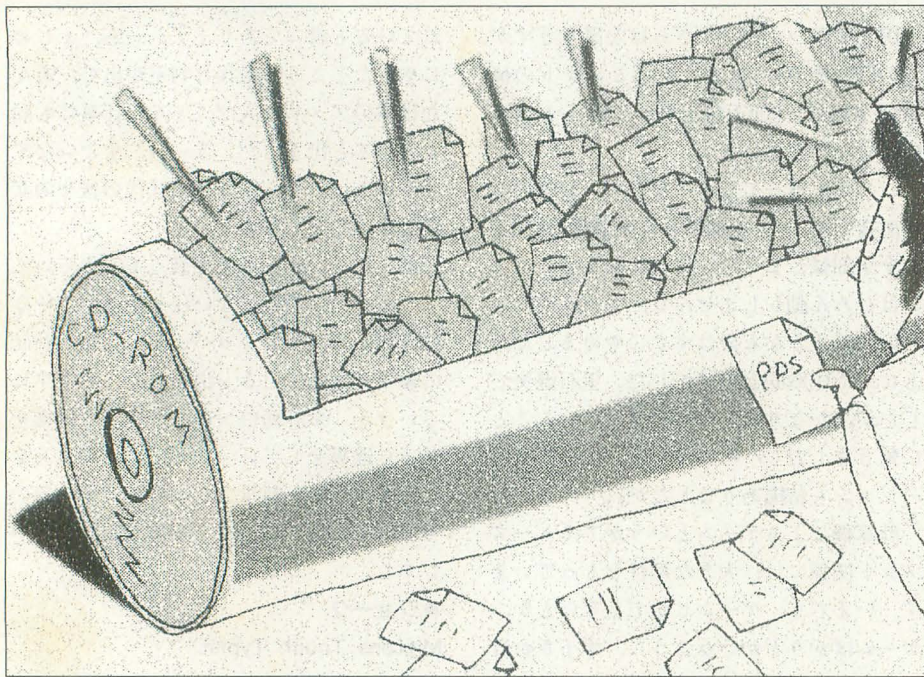
Mittens Touch Typist

これは悪質ないたずらといえます。これをシステムフォルダに入れて立ち上げると，正しくキー入力してもしょっちゅう間違えて表示されてしまうというものです。プログラム名のとおり，ミトン手袋をしてタイピングをするようなものというわけです。

もし，これを増殖するようにしたならば，完全なコンピュータウィルスであり，犯罪になるでしょう。しかし，このプログラムは，1回走ると自分で自分を壊すということなので，一過性の害ですみます。そこはきちんと明確な線引きをしているのですね。

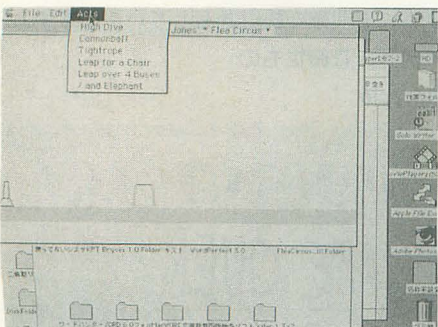
●流行・ご時世もの

Recycle

ベルトコンベアで流れてくるいろいろなゴミをビン，生ゴミ，金属，化学薬品などにおお急ぎで分類していくアクションゲームです。さらにトランプカードやはさみ，

玉石混交の山の中で輝く妙なソフト

指輪などが流れてきてボーナスポイントもあり，娯楽性も高めてあります。

ゲーム説明の最後のところで，本当はこんな大ざっぱな分類ではだめでもっと細かく分類しなければならないのだよと大真面目に説明しているところが笑えます。エコロジー便乗ゲームですね。まあまあ楽しめます。

●ミョーなソフト

FleaCircus

画面に小さな小さなのみが出てきて，ぴょんぴょんはねてはいくつかの芸をするというものです。多くの芸で失敗して死んでしまいます。物静かでシュールで，それでいてユーモアが感じられるミョーなソフトの典型でしょう。

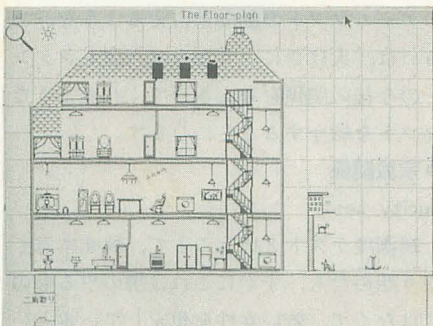
お金を送るともっともっとミョーなソフトを送ってくれるそうです。

●いかにもアメリカ

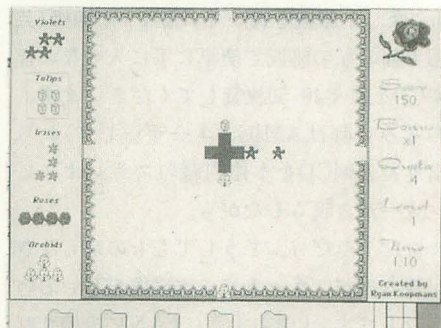
PhrazeCrazePlus

これは最初のマックが出て間もないころに作られた,英単語(といっても複数の単語からなるフレーズが主)を当てるゲームです。しかし，そこはクイズ関係のテレビ番組が異常に発達したアメリカのことですから，得点の加算にルーレットを使ったり，司会者が出てきたり，効果音を使ったりして，娯楽性を盛り上げています。

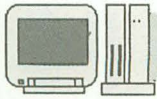
●文句なしに推薦

dollshouse_pro

いわゆるドールハウスを画面上に作って楽しむソフトです。子供にもウケるでしょう。まず，天井や壁や階段などを配置することにより建物を作り，電灯をつけ，家具を配置し，犬や猫をおいていきます。各種パーツは最初から登録してあるので，マウスをクリックするだけで難しいことはまったくありません。

テレビ，シャワー，犬のシッポ，煙，テレビなどのパーツはたえずアニメ的に小さく動いています。さらに，作成モードからブラウズモードに変えて，マウスでそれらのパーツをクリックすると，さらにいろいろな動きや音が楽しめます。

Chrysanthemum

最後は，なかなかの完成度をもち，しかも皆さんがのめりこみそうな感じのゲームを紹介しましょう。日本人が作ったというわけではないのですが，生け花を題材とした日本情緒(?)あふれるゲームです。ただし，始まりの音楽はなぜかツインピークスです。

ゲームが始まると，上下左右から小さな花がひとつずつ降ってきます。そして障害物があるとそこにぶつかって止まります。この花の種類によってパターンをいくつか構成すると面クリアとなります。時間制限があり，最初はすぐにゲームオーバーになります。しかし，熟練度に応じてきちんと点数が伸びていきそうな感じです。

このソフトは気に入ったらお金をちょうだいというタイプのPDSではなく，ジョブウェアと称しています。要するに，気に入ったら僕にお仕事ちょうだいというわけです。日本人の作った漢字変換のPDSで，使うようになったら，こういう団体のどれかに募金してください，というのもありましたが，作った人の個性が出ていてなかなか面白いですね。

仮想ドライバの開発実験PART5.

仮想ドライバの改良

電机本舗　由井 清人　Yui Kiyoto

今回は，7月号に掲載したR.Cを改良して，ドライブ認識機能を追加します。PC-9801，DOS/Vマシンのドライブをどうやってシステムに認識させるか，本文，そしてコラムを参照しながらその手法を学んでください。

今回は，前回まで作ってきた仮想ドライバを小改良してみます。前回のレポートで，当システムの特性(欠点？)がだいぶ見えてきました。それらを踏まえて，直すことができるかどうかチャレンジしてみたのが今回のレポートです。ただ，いろいろなところで障害が起きてしまいあまり多くを直すことはできませんでした。主に，起動時の仮想ドライバの認識制御を作り替えました。変更はすべて，従機側の制御プログラム「R.C」です。システムの使用方法，コンパイル方法は従来と同じです。

今回の改良点

これまでのシステムですと，起動時には仮想ドライブにフロッピーディスクがセットされていなければなりませんでした（ハードディスクのときはこのかぎりではありません）。

これは，起動したときに仮想ドライブに指定したディスクを読み，そこで初めてディスクの種類を識別するからです。主機はこの識別データに従いデバイス登録を行うわけです。

この形式ですと，毎回フロッピーディスクをセットしておかなければならないので都合がよくありません。それに，指定のドライブが1基ならば問題ないのですが，複数の場合は大変です。このようなわけでなんとかならないかとアルゴリズムを見直してみました。実は筆者の想定している使い方というのは，X68000を主機として，従機にPC-9801ないしDOS/Vを接続するというものです。PC-9801にしろDOS/Vにしろ，フロッピーディスクは通常2DDと2HD (DOS/Vであれば2HC) の2種類あります。ですから，アルゴリズムを見直して，あわよくば，2DDと2HD (2HC) の自動識別ができるのではないかということも含め着手してみました。

デバイスドライバ登録のメカニズム

少しおさらいになりますが，このシステムは図1のような形で立ち上がります。このときに，主機側での制御をもう少し覗いてみると図2にあるフローチャートのようになります。ここで，主機は従機と交信，結果，デバイスドライバの登録データを取得しHuman68kに設定登録を行います。

図1

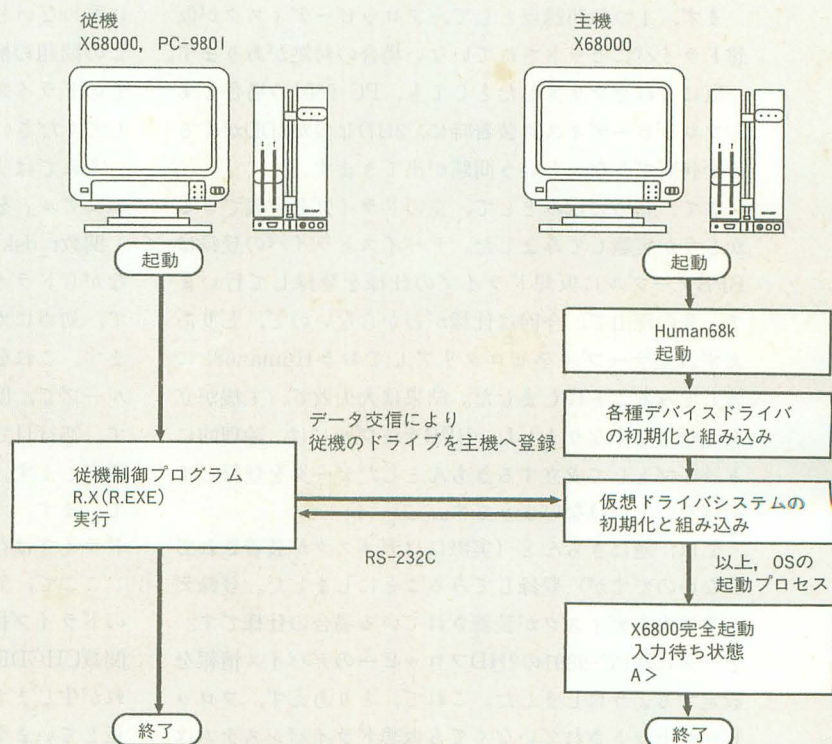
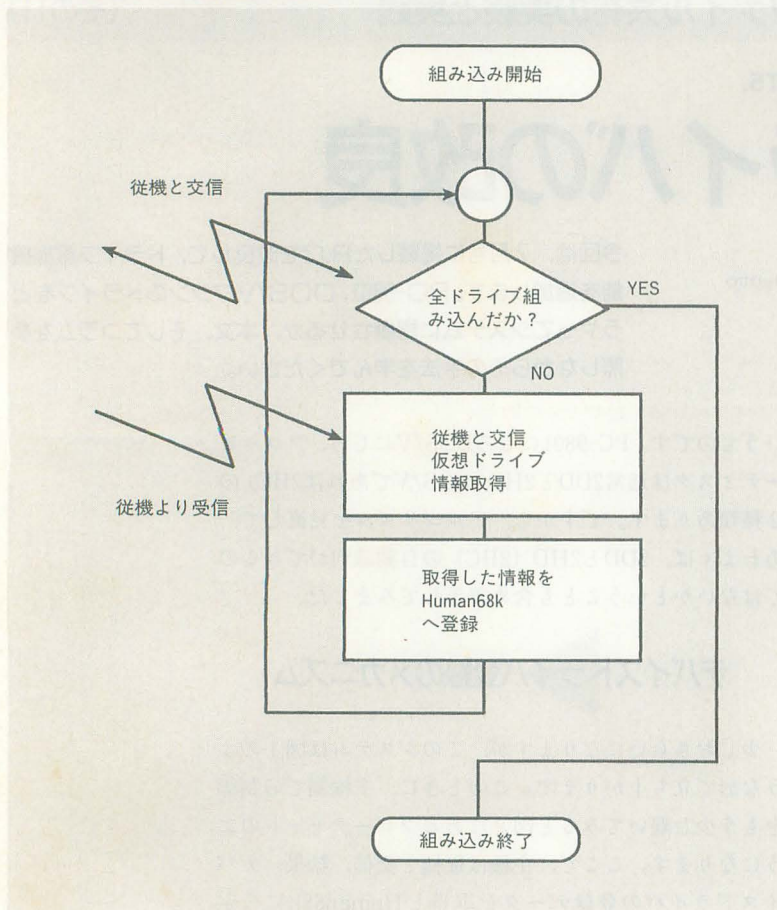



図2　仮想ドライバ組み込みの流れ

仮登録による問題先送り

まず，1つの問題点として，フロッピーディスクが仮想ドライバにセットされていない場合の対処があります。

仮にこれをクリアしたとしても，PC-9801の場合ですとフロッピーディスク装着時に，2HDなのか2DDがくるのか判断できないという問題が出てきます。

さて，初めに試みとして，空のドライブを定義できるかどうか実験してみました。デバイスドライバの登録はBPBテーブルに仮想ドライブの仕様を登録して行います。先の理由で，今回は仕様がわからないので，とりあえず設定テーブルをゼロクリアしておきHuman68kに渡してみることにしました。結果は大失敗で，主機が立ち上がらなくなりました。BPBテーブルには，論理的にドライブとして成立するきちんとしたデータを登録してあげないといけないようです。

では，逆にきちんと（実際にはディスクが装着されていないのですが）登録してみることにしました。登録データは本来ディスクが装着されている場合の仕様です。データにはPC-9801の2HDフロッピーのデバイス情報を設定するようにしました。これで，とりあえず，フロッピーがセットされていなくても仮想ドライバシステムは立ち上がるようになります。

仮登録の実際

具体的にはリスト1を参照してください。ここには関数r_dskini()を示します。前回までのr_dskini()と置き換えてください（7月号掲載のリスト2，ソースコードR.C中の552～614行目を参照，ここを差し替えます）。

関数r_dskini()は，ブロック型デバイスドライバの初期化プログラムです。機能は，Human68kのBPBテーブルにドライブ情報を格納して受け渡すことです。r_dskini()は，従機側プログラムですので，仮想ドライブのBPBテーブルを作成し，主機側のほうへRS-232Cにて送出する役目を担っています。

おおまかには，このようなものですが，実際には次の基本機能をもっています。

1）仮想化するドライブの数を把握する
2）仮想化するドライブにディスクが装着されているかどうか識別する。ディスクが存在すれば，そのディスクのBPBデータを取得し設定する。もし，存在しなければ，仮設定として，2HDのBPBを設定する
3）BPBデータを仮想化するドライブの数だけ主機に送信する

特に1）の項目が技術的問題となってきます。従来は，えいやあとBPBを取得するためのシステム関数を呼べば，OSがディスクアクセスしてくれました。ディスクが存在しなければ，エラーとなりユーザーがディスクをセットするまで中断するだけでよかったのです。しかし，今回は，そうはいきません。処理を中断することなく先に進めないと主機の立ち上がりが途中で止まるからです。この問題の解決方法の詳細はコラム「Human68k/DOSでのドライブ数の調べ方」に書いてありますので，参照してください。

それではリスト1です。以後，XCの「Cライブラリマニュアル」を脇に置きながら読んでください。

関数r_dskini()は内部で，カレントドライブを移動しながらドライブをチェックする処理をします。したがって，初めにカレントドライブを保持しておく必要があります。これを21行目で行っています。23行目から始まるループで，仮想化するドライブのチェックを行っています。25行目で次にチェックしたいドライブをカレントに指定します。続く26行目で，カレントドライブを取得しています。カレントと指定ドライブが等しければ，そのドライブは存在することになります。

ここで，気をつけなければいけないのは，当システムのドライブ指定は1＝A:の扱いになっているのに対し，関数CHGDRV()，CURDRV()は0＝A:となっており，ずれが生じます。これを25，27行目で値の加減を行って補正しています。

次に29行目で，指定ドライブが存在するかの判定を行

っています。もし，なければ，30〜32行のブロックで，仮想ドライブ登録の終了を主機に宣言して，このループより抜けだして終了します。

36行目のDOS関数DRVCTRL()で，このドライブにディスクが装着されているかのチェックを行っています。返り値の0〜31ビット中，第1ビットがディスクの装着を示します。ですから，37行目で2で論理積（アンド）をとり，1ビット目だけを抽出しています。このビットが立っているときに，ディスクは装着されています。

43〜55行目が装着されていないときの処理です。PC-9801の2HDのディスクデータを主機に送るパラメータに設定しています。

反対に装着されていれば，58〜70行目を実行します。ここでは，ディスクを読んでDPBテーブルを取得し，主機へ送るパラメータを設定しています。

75〜93行目までで，主機へ仮想ドライブの設定情報を送信しています。これらの処理を，ドライブの最大値まで繰り返し行います。最後に，95行目で，カレントドライブを元に戻して終了します。

FILE

仮登録されたドライブの動作

仮登録されたディスクは，主機に仮想ドライブとして登録され，使用できるようになります。このとき，注意してほしいのは，2HDディスクドライバとして登録されているということです。つまり，以後は2HDディスクし

Human68k/DOSでのドライブ数の調べ方

Human68k(DOS)において，このようなソフトを作るときの欠点として，標準のシステムコールの中にドライブの数を調べる機能がないという点があります。本文で行っているような，仮想ドライブシステムですと，的確にドライブ数を把握しないと根本的に動くものができません。ここでは，変則的なシステムコールによる実現を紹介します。

まず，本当にドライブ数をチェックする方法がないのでしょうか。XCのDOS関数CHGDRV()というのがあります。この関数の要約は次のようなものです。

●int CHGDRV(int drive)
機能：driveで指定したドライブをカレントにする。driveの値は0(A:)，1(B:)……のように指定する
戻り値：最大論理ドライブ数

使い方は次のようなものです。

```
        :
/* カレントドライブをdrvの示す値にして
   max_drvに最大論理ドライブ数を格納 */
    max_drv = CHGDRV( drv );
        :
```

なんだ，ドライブ数がわかるではないかと，皆さんは思うでしょう。

しかし，ここで返ってくる値は論理ドライブ数と呼ばれる得体の知れないもので，実際のドライブ数とはまるで別ものです。

論理ドライブ数とは，Human68k(DOS)に接続できるドライブの数を表しているのです。つまり，現在つながっているドライブの数ではなく，最大いくつまでディスクを増設できるかの値でしかないのです。

これは，AUTOEXEC.BATの中に「LASTDRIVE=××」の指定があったことを思い出してください。ここと連動するものなのです。

したがって，ここで必要とする接続されているドライブ数ではありません。そして，DOSコールを調べる限りは，これ以外にそれらしい機能は存在しません。

観点を変えて，ハードに依存する関数ならばどうでしょうか。確かに，IOCSコールの中に，B_DRVSNS()が存在します。

●int B_DRVSNS(int drive)
機能：driveで指定したドライブをカレントにする。driveの値はint型4バイト16進でいう8桁$0000xx00の中で，××で表した桁にドライブを表す値をセットする。××には次の値をセット

ハードディスク＝0x80〜0x8f
フロッピーディスク＝0x90〜0x93

戻り値：ドライブが存在するか否かをステータスにて返す

確かに，この命令を使えばディスクの最大数を調べられそうです。ループを作り，各ドライブを指定して存在するかどうか返り値で判定すればいいわけです。存在するドライブをひとつずつ数えていけば最後には最大ドライブ数が求められるはずです。

しかし，実際には，この手法は使えません。もしくは，あまりうまい方法ではありません。それは，次のような理由によります。

1) 機種依存性が強すぎる（今回の場合は特にPC-9801への同時移植を行っている）
2) 接続しているドライブをHuman68kが認識しているかどうか不明。また，1つのディスクでパーテションを複数切った場合どういう扱いになるか検証が必要
3) フロッピーとハードディスクの異なるパラメータを与えて集計するので一元化されていない

特に，3)が大きな不安要素となります。1つのループで，A:，B:，C:……のように一元的にチェックできないと不安を感じます。MOや，CDのようなデバイスは，この手法で検出できるか不明です。

この手法は，現在接続されているFDDの数を検出するとき，またHDDの検出のときに有効であり，OSに登録されている総ドライブ数を検出するには向かないと思います。

では，どうするか。少なくとも，A:，B:，C:……というように，OSのドライブ番号で具体的に指定し，存在するか否かチェックできる方式でないといけません。

解法のひとつとして，再び，関数CHGDRV()を利用する方法が考えられます。CHGDRV()は本来，カレントドライブを指定するものです。

ここで，CHGDRV()に存在しないドライブを指定した場合を思い浮かべてください。当然エラーになるはずです。OS上に登録されていない，存在していないドライブを指定したところで機能のしようがないからです。

これならば，うまくいきそうです。ただ，CHGDRV()はエラーを返さないので，ドライブの変更が成功したか失敗したか判定することができません。ここでなにか別の工夫をして判定する必要があります。

ここでいったん整理してみましょう。

CHGDRV()を実行すると，次のいずれかの結果を招くはずです。

・成功……カレントドライブが指定ドライブになる
・失敗………カレントドライブが指定ドライブにならない

つまり，CHGDRV()を実行後，指定どおりにドライブが変化したかを見ればいいわけです。これをもとに最大ドライブ数を得る関数をサンプルとして示します。

```
int get_maxdrv()
{
  int i;
  int j;
      for( i=0 ; i<26 ; i++ ) {
    CHGDRV( i );  /*ドライブを変更*/
    j = CURDRV(); /*現在のドライブを取得 */
               /*現在のドライブは指定どおりか*/
      if( i != j ) {
          /*NO= 存在しないドライブを
                         指定した．終了*/
        break;
      }
      }
  return( i )
```

簡単なパズルです。ちなみに，このサンプルには致命的な欠点があります。それは，これを実行すると，カレントドライブが最大ドライブに移動するということです。本来ならばきちんとカレントドライブを戻す必要があります。まあ，サンプルですからこれもご愛嬌です。

か読めなくなるということです。

X68000ですと，フロッピーディスクは2HDしか標準で読めないので特に問題はありません。ですが，PC-9801などは2DDも読めるので不都合が出てくると思います。このようなときには，仮登録ではなく，初めから仮想ドライブに2DDを装着しておいてください。そうすれば，主機へは2DDドライブとして登録される形になります。

当初は，2DD/2HDの切り替えを動的に行えるかもしれないと思い，アルゴリズムを検討してみました。ですが，いろいろ動作を追跡してみると，デバイスドライバの初期化を行うr_init()が根本的に起動時の1回しか呼ばれないので断念しました。仮想ドライバはr_init()にドライバ情報を設定しHuman68kへ受け渡すのです。これが，以後呼ばれないのであればお手上げ状態です。

主機側の仮想ドライブのBPBテーブル格納領域はシステムに常駐しています。ですから，ここを勝手に書き換えれば，2HD/2DDの動的変更をできるかもしれません。ですが，Human68kを無視してその管理領域を書き換えるリスクを考え，躊躇しました。加えて，実際の運用を考えたとき，登録時にフロッピーを明示的に入れておけば2DDでも読めるわけです。

また，今回はつけませんでしたが，起動時オプションで，仮登録のディスクタイプを2DDに指定する形にすれば構造がすっきりするし，運用上問題ないとのことより，動的管理へのアプローチをあきらめました。

仮登録されたドライブへのアクセス

さて，このようにして，ディスクが装着されていなくても主機より利用できるようになりました。利用すると当然，アクセスが発生します。このときに，仮想ドライブにディスクが装着されている必要があります。もし，セットされていなければ，フロッピーをセットするようにメッセージを画面に出して，処理を一時中断，ユーザーがフロッピーを装着後に処理を再開すればいいわけです。

7月号掲載のソースコードR.Cでは，ディスクの読み書きをDISKRED()およびDISKWRT()の低水準DOS関数で行っています。これらの関数は都合の悪いことに，エラーステータスを返しません。動作が成功したかどうかを知る術がないのです。この関数をそのまま使用しているかぎりは，ディスクが未装着のときの処理ができません。

したがって，新しい上位互換関数DISKREDX()およびDISKWRTX()を作り，差し替えることにしました。新しい2つの関数は，ディスクがセットされていないと従機側の画面にディスクの装着を促すメッセージとビープ音を鳴らし，処理を中断します。ユーザーは，ディスクをセットしてから，任意のキー入力により処理を継続します。具体的なプログラムはX68000とPC-9801で異なるので次章にゆずります。

ただ，7月号掲載のソースコードR.C中，DISKRED()およびDISKWRT()している行を次のように変更してください。単に使用している関数名を変更するだけのことです。

●変更前

```
423:  DISKRED( (unsigned char *)_rw_buf, req->
unitcd+_drv, rec, wk1 );
                    ⋮
487:  DISKWRT( (unsigned char *)_rw_buf, req->
unitcd+_drv, rec, wk );
                    ⋮
```

●変更後

```
423:  DISKREDX( (unsigned char *)_rw_buf, req
->unitcd+_drv, rec, wk1 );
                    ⋮
487:  DISKWRTX( (unsigned char *)_rw_buf, req
->unitcd+_drv, rec, wk );
                    ⋮
```

DOSでのディスク装着の確認方法

本文にてX68000固有のDOS関数DRVCTRL()を使用しています。これは，指定したドライブの状況をチェックし，きめ細かく調査する機能です。本文では，ディスクが装着されているかどうかの判定に利用しています。

この関数，一応DOS関数（MS-DOS互換）の1つなのですが，ここだけ，互換性がなく該当する機能がPC-9801にはありませんでした。

困ったことにざっと見渡した限りでは，該当する機能がありません。コラム「Human68k/DOSでのドライブ数の調べ方」でも少し触れていますが，BIOS（X68000でのIOCSに相当）にドライブのチェック機能があるのですが簡単には使えません。使えたとしても，BIOS指定のドライブがMS-DOSのどの番号に対応しているか調べるのが大変です。

まじめなチェックをあきらめ，指定ドライブをリードしてみて読めるか否かという変則的な方法を取りました。これがリスト4の関数DRVCTRL()です。

ひと口にディスクを読めばいいといってもなかなか厄介です。通常のDOSのディスクリード機能を使用すると，MS-DOSレベルでエラーメッセージを出し，制御が戻ってきません。このときのメッセージはいわゆる，

DRIVEの準備ができていません．
中止<A>，もう一度<R>，無視<I>?

という例のメッセージです。これが出ると，ユーザーがなにかしらキー入力しないと先に進まないので非常に都合が悪いわけです。

結局，DOSの機能の中でも最も低水準の，いわばDOSファンクションのさらに下の階層のリード機能を使って実現しました。俗にいうint25hのダイレクトディスクリード機能を使用しました。この機能は，読み取りを行った結果が，8086のALレジスタに格納されます。ALがゼロのときに正常リード，ALが有値のときはエラーです。特に第1ビット（0～15ビットのうち）が立っているときはディスク未装着です。

ですから，この機能を使って調べたいディスクに読み取りをかけ，ALレジスタの第1ビットを返すようにして作りました。

X68000での変更点

この場合は，ただ単にDISKREDX()およびDISKWRTX()を新たに作り，ソースコードR.Cに追加してコンパイルし直すだけです。追加分をリスト2に示します。

非常に簡単なプログラムです。DISKREDX()もDISKWRTX()も中でそれぞれ，従来のDISKRED()とDISKWRT()を呼んでいるだけです。ただ，この両者は，エラーチェック機能がないので，それぞれ前にDRVCTRL()を呼び，ディスクが装着されているか否かを判定しています。

装着されていれば，そのまま読み書きを行います。未装着のときには，ディスクを装着するように画面にメッセージを出し，キー入力を促すようにしています。

PC-9801のときの変更点

ここまで説明してきたのは，X68000上の従機プログラムR.Cでの変更点でした。PC-9801の場合は，さらにもう少し変更が必要です。

まず，これまでの変更でいくつかX68000固有の関数を使用しました。すなわち，CHGDRV()，CURDRV()，DRVCTRL()の3本です。

このうち，CHGDRV()，CURDRV()は今回初めて利用する関数ですので，PC-9801用互換関数を新規に作ってあげます（リスト3）。これをR.Cの最後にでも追加すればよいでしょう。幸いCHGDRV()，CURDRV()は，DOS関数ですので，MS-C上にもほとんど同じ関数が存在しています。

_dos_get drive()と_dos_set drive()がそうです。機能はほとんど同一です。ただ，パラメータが，X68000の場合，0(＝A:)から始まるのに対して，1(＝A:)である点が注意が必要です。パラメータを適当に加減すればそのまま利用できます。

差し替え変更点

リスト4に在来の関数の変更を示します。変更の対象は次の関数です。

1) DISKRED()
　注　今回より，DISKREDX()に改名

2) DISKWRT()
　注　今回より，DISKWRTX()に改名

3) DRVCTRL()

DISKRED()とDISKWRT()は前章で説明したとおりの変更です。対して，DRVCTRL()は従来X68000と互換性を取るためだけのダミーだったのですが，今回はきち

メディア交換チェックの怪

ブロック型デバイスドライバの中に，メディアが交換されたかどうかを調べるプログラムがあります。当システムも例外ではなく，「r_mediac()」という関数名で存在します。Human68kが適時においてフロッピーの抜き差しが行われたかどうか知りたいときに，この関数を呼び出して，メディア交換の状態のチェックを行うものです。

当システムでは，この関数を橋渡しにしてHuman68kへ交換情報を返すわけです。リスト5にr_mediac()の抜粋を示します。非常に簡単なプログラムです。

実際に交換情報を設定しているのは次の箇所です。

```
10:    req_chg->diskfg = 0L;
```

この10行目が，ちょうど当システムにおけるパラメータの受け渡し設定を行っているところです。行中の構造体変数「req_chg->diskfg」がディスクを交換したかどうかの値を格納するものです。本来ここは2値を取り，

　ディスクは交換されていない＝1
　ディスクは交換された＝－1

のいずれかを指定する規則になっています。

しかし，ここでは無条件にゼロを指定しています。これは，当システムの原型となったXCの「プログラマーズマニュアル」記載のSRAMディスクドライバの中で，この値を設定しているためです。

SRAMディスクは当然，着脱不可能です。したがって順当に考えると，1＝固定（ディスクは交換されていない）となります。ただ，実際に動いているプログラムでゼロを指定しているので，これを優性としました。

ただ，連載の過程でSRAMディスクを前提とした仮想ドライブからFDDおよびHDDにいたる汎用に変化した現在では，不適当です。

少し乱暴ですが，ここには強制的に－1，すなわち，ディスクが交換されたことを示す値を常に設定すべきです。プログラマーズマニュアルに従うならば，ディスクが交換されたかどうか不明な場合は常に－1を設定するように推奨しています。

また，MS-DOSのフロッピーディスクですと，通常は手動でディスクを交換し，カンチレバーを回すことになりますから根本的にディスクの交換を識別できません。ですから，MS-DOSのフロッピーディスクを仮想ドライブとして利用する場合を考えると，ここの値は常に－1固定が好ましいことになります。

もっとも，これはDOSとの接続を考えるからこうなるので，X68000同士であれば，オートイジェクトを標準装備していますので，プログラムで，ディスク交換されたか否かきちんと検出できるはずです。いずれにせよ，ここではフェイル・セーフ（障害が起きたときに安全なほうへ倒れるように配慮すること）の原則に従い，－1に設定してみました。

しかし，結果は惨敗でした。それまで正常に動いていた当システムがここを－1に変更しただけで動作しなくなりました。

ディスクのリードは正常に作動するのですが，ライト処理がおかしくなるようです。ライトを伴うディスク操作をすると暴走するようになりました。

MS-DOSの資料を見ますと，ブロック型デバイスドライバのメディア交換チェック（コマンドコード1）を呼び出して，もし，メディア交換がされているようならば，BPBテーブルの作り直し命令（コマンドコード2）をDOSがコールして，ドライブ情報を設定し直すということです。

ここで，はたと思い当たりました。まず，当システムでは，コマンドコード2は作っていなかったのです。なければ動くはずもなく，おかしくなって当たり前ですね。さらに，もしやと思い，XCの「プログラマーズマニュアル」（ただしver2.0）を見ると，そもそもコマンドコード2は未使用ということでした。どうやら，この資料の時点（おそらくHuman68k ver.2.0時代）の状況では未サポートである，ということなのでしょう。

んと中身を作りました。

このリスト4を7月号掲載のソースコードR.C中の734〜794行と差し替えてください。

ここの説明は不要でしょう。ただ，ディスクの制御&チェックルーチンであるDRVCTRL()にだけは触れておきます。いろいろ調べたのですが，DOSレベルにはディスクのチェック機能が用意されていないようです。しようがないので，チェックするにあたり，低水準のディスクの読み書き機能（X68000と異なります）がきちんとエラーチェックしていることに着目し，適当なデータを実際に読んでみて，エラーになるか，正常かで判定をするようにしました。プログラムの移植を考えている方の参考になればと思い，述べておきます．

最後に

今回はひどくマイナーな実験レポートになってしまいました。さて，次回は，まだ流動的ですが，DOSのバージョンによる動作検証を行いたいと思います。DOSも，ver.3.1，3.3，および5.×，6.×系と上がってきています。フロッピーだけならば問題ないのですが，ハードディスクですと仕様が相当変化しているおそれがあります。

DOSのパッケージを用意するだけでかなり労力を必要としそうな気がしますが，いずれはクリアせねばならない問題であるはずです。地味ですがやらないわけにはいきませんからね。

リスト1

```
 1: /*****************************************************/
 2: /*    dskini    ディスク初期化ルーチン             */
 3: /*****************************************************/
 4: int r_dskini( req )
 5: struct REQ_INI *req;
 6: {
 7:    int    sts;
 8:    struct   DPBPTR   d;
 9:    struct   BPB_TBL    bpb_tbl;
10:    int    fat;
11:    int    fat_no;
12:    int    rsv_sct;
13:    char    dsk_flg;
14:    int    drv;
15:    char    d_no;
16:    long    mode;
17:    long    old_drv;
18:    long    wk_drv;
19:
20:     drv = _drv;
21:     old_drv = CURDRV();        /* 現在のドライブ保存    */
22:
23:     while( 1 ) {
24:
25:         CHGDRV( (long)drv-1 );
26:         wk_drv = CURDRV();
27:         wk_drv++;
28:
29:         if( drv != wk_drv ) {
30:            dsk_flg = -1;
31:            sts = blk_out( &dsk_flg, sizeof(dsk_flg) );
32:            break;
33:         }
34:
35:         mode = 0L;
36:         mode = DRVCTRL( mode, drv );
37:         mode &= 2L;
38:
39:         d_no = 'A' + (char)drv;
40:         d_no --;
41:
42:         if( mode==0 ) {
43:            bpb_tbl.b_no = 1024;        /* セクタあたりのバイト数 */
44:            bpb_tbl.sct_no = 1;         /* クラスタあたりのセクタ数 */
45:            bpb_tbl.fat_no = 2;         /* ファット領域の個数 */
46:
47:            bpb_tbl.rsv_sct_no = 1;     /* 予約領域のセクタ数 */
48:            bpb_tbl.root_ent_no = 192;  /* ルートの最大ファイル数 */
49:            bpb_tbl.sct_max = 1222;     /* 全セクター数 */
50:            bpb_tbl.id = 1;             /* メディアバイト */
51:            bpb_tbl.fat_sct_no = 2;     /* 1fatのセクタ数 */
52:
53:            d.byte = 1024;
54:
55:            printf( "%c: を主機へ仮想ドライブとして登録. DISKが未セットなので2HD
に設定. ¥n", d_no  );
56:         }
57:         else {
58:          sts = GETDPB( drv, &d );
59:
60:          bpb_tbl.b_no = d.byte;          /* セクタあたりのバイト数 */
61:          bpb_tbl.sct_no = d.sec + 1;   /* クラスタあたりのセクタ数 */
62:          bpb_tbl.fat_no = d.fatcount; /* ファット領域の個数 */
63:
64:          bpb_tbl.rsv_sct_no = d.fatsec;     /* 予約領域のセクタ数 */
65:          bpb_tbl.root_ent_no = d.dircount; /* ルートの最大ファイル数*/
66:          bpb_tbl.sct_max = d.maxfat * (d.sec+1); /* 全セクタ数 */
67:          bpb_tbl.id = d.id;                /* メディアバイト */
68:          bpb_tbl.fat_sct_no = d.fatlen; /* 1fatのセクタ数 */
69:
70:             printf( "%c: を主機へ仮想ドライブとして登録 ¥n", d_no );
71:         }
72:
73:         dsk_flg = 0;
74:
75:      /* bpb tbl 送信    */
76:         if( (sts=blk_out( &dsk_flg, sizeof(dsk_flg) ))) {
77:            break;
78:         }
79:         else {
80:            XCHG2( &(bpb_tbl.b_no) );            /* もしDOSなら配列変換 */
81:            XCHG2( &(bpb_tbl.rsv_sct_no) );   /* もしDOSなら配列変換 */
82:            XCHG2( &(bpb_tbl.root_ent_no) ); /* もしDOSなら配列変換 */
83            XCHG2( &(bpb_tbl.sct_max) );        /* もしDOSなら配列変換 */
84
85:     /* bpb tbl 送信    */
86:            if( (sts=blk_out( &bpb_tbl, sizeof( bpb_tbl ) ))) {
87:               break;
88:            }
89:
90:            _byte[drv-_drv] = d.byte;
91:
92:            drv ++;
93:         }
94:     }
95:
96:     CHGDRV( old_drv );              /* ドライブ復旧 */
97:
98:     return( sts );
99: }
```

リスト2

```
 1:
 2: /*****************************************************/
 3: /*    DISKREDX    ディスクリードルーチン            */
 4: /*****************************************************/
 5: void DISKREDX(
 6:     unsigned char* buf, int drv, long rec, long len )
 7: {
 8:    unsigned int    sts;
 9:    int    i;
10:    long    mode;
11:
12:     while( 1 ) {
13:        mode = 0L;
14:        mode = DRVCTRL( mode, drv );
15:        mode &= 2L;
16:
17:        if( mode ) {
18:           DISKRED( buf, drv, rec, len );
19:           break;
20:        }
21:
22:        printf( "¥aディスクをセットして、どれかキーを押してください。" );
23:
24:        getch();
25:        printf( "¥n" );
26:     }
27: }
28: /*******************************************************/
29: /*    DISKWRTX    ディスクライトルーチン              */
30: /*******************************************************/
31: void DISKWRTX(
32:    unsigned char* buf, int drv, long rec, long len )
33: {
34:    unsigned int    sts;
35:    int    i;
36:    long    mode;
37:
38:
39:     while( 1 ) {
40:        mode = 0L;
41:        mode = DRVCTRL( mode, drv );
42:        mode &= 2L;
43:
44:        if( mode ) {
45:           DISKWRT( buf, drv, rec, len );
46:           break;
47:        }
48:
```

▶ようやくハードディスクを手に入れた。これで少しは人並みの生活ができると思ったが，「スパII」の広告を見ると，まだまだという気分になるが，もう金ねーぞ！

平　勝久(20)大阪府

```
49:          printf( "¥aディスクをセットして、どれかキーを押してください。" );
50:
51:          getch();
52:          printf( "¥n" );
53:      }
54: }
55:
```

リスト3

```
 1:
 2: /*****************************************************/
 3: /*   CURDRV   現在のドライブ取得ルーチン           */
 4: /*****************************************************/
 5: long CURDRV()
 6: {
 7:   unsigned int   drv;
 8:
 9:    _dos_getdrive( &drv );
10:    drv--;
11:
12:    return( (long)drv );
13: }
14:
15:
16: /*****************************************************/
17: /*   CHGDRV   現在のドライブ変更ルーチン           */
18: /*****************************************************/
19: long CHGDRV( long drv )
20: {
21:   unsigned int   drv_max;
22:
23:    drv++;
24:    _dos_setdrive( (int)drv, &drv_max );
25:
26:    return( (int)drv_max );
27: }
28:
```

リスト4

```
 1: /*****************************************************/
 2: /*   DISKREDX    ディスクリードルーチン           */
 3: /*****************************************************/
 4: void DISKREDX(
 5:   unsigned char* buf, int drv, long rec, long len )
 6: {
 7:   unsigned int    sts;
 8:   int    i;
 9:
10:    for( i=0 ; i<5 ; i++ ) {
11:       sts = diskred( buf, drv, rec, len );
12:       if( sts==0 ) {
13:          break;
14:       }
15:       else if( sts==2 ) {
16:          printf( "¥aディスクをセットして、どれかキーを押してください。" );
17:       }
18:       else {
19:          printf( "¥aディスクに障害があります。確認後、どれかキーを押してください。
" );
20:       }
21:
22:       getch();
23:       printf( "¥n" );
24:    }
25: }
26:
27:
28: /*******************************************/
29: /*      diskred ディスクリードルーチン   */
30: /*******************************************/
31: int diskred(
32:   unsigned char* buf, int drv, long rec, long len )
33: {
34:   union   REGS    i_reg;
35:   union REGS    o_reg;
36:   struct  SREGS    s_reg;
37:   unsigned int    sts;
38:   unsigned int    ds;
39:
40:    drv --;
41:    segread( &s_reg );
42:    ds = s_reg.ds;
43:
44:    i_reg.h.al = drv;
45:    i_reg.x.bx = (unsigned int)buf;
46:    i_reg.x.cx = len;
47:    i_reg.x.dx = rec;
48:
49:    int86x( 0x25, &i_reg, &o_reg, &s_reg );
50:    dlytime();
51:
52:    sts = o_reg.x.ax;
53:    sts &= 0x000f;
54:
55:    return( sts );
56: }
57:
58: /*******************************************/
59: /*   DISKWRTX    ディスクライトルーチン   */
60: /*******************************************/
61: void DISKWRTX(
62:   unsigned char* buf, int drv, long rec, long len )
63: {
64:   unsigned int   sts;
65:   int   i;
66:
67:    for( i=0 ; i<5 ; i++ ) {
68:       sts = diskwrt( buf, drv, rec, len );
69:       if( sts==0 ) {
70:          break;
71:       }
72:       else if( sts==2 ) {
73:          printf( "¥aディスクをセットして、どれかキーを押してください。" );
74:       }
75:       else {
76:          printf( "¥aディスクに障害があります。確認後、どれかキーを押してください。
" );
77:       ;
78:
79:       getch();
80:       printf( "¥n" );
81:    }
82: }
83:
84:
85: /*****************************************/
86: /*      diskwrt ディスクライトルーチン   */
87: /*****************************************/
88: int diskwrt(
89:   unsigned char* buf, int drv, long rec, long len )
90: {
91:   union   REGS    i_reg;
92:   union REGS    o_reg;
93:   struct  SREGS    s_reg;
94:   unsigned int    sts;
95:   unsigned int    ds;
96:
97:    drv --;
98:    segread( &s_reg );
99:    ds = s_reg.ds;
100:
101:    i_reg.h.al = drv;
102:    i_reg.x.bx = (unsigned int)buf;
103:    i_reg.x.cx = len;
104:    i_reg.x.dx = rec;
105:
106:    int86x( 0x26, &i_reg, &o_reg, &s_reg );
107:    dlytime();
108:    sts = o_reg.x.ax;
109:    sts &= 0x000f;
110:
111:    return( sts );
112: }
113:
114:
115:
116: char _wk[8096];
117: /*****************************************************/
118: /*    DRVCTRL     ドライブ情報取得ルーチン          */
119: /*****************************************************/
120: long DRVCTRL( long mode, int drv )
121: {
122:   union    REGS    i_reg;
123:   union REGS    o_reg;
124:   unsigned int    sts;
125:
126:    i_reg.h.ah = 0x1c;
127:    i_reg.h.al = drv-1;
128:    i_reg.x.bx = (unsigned int)(_wk);
129:    i_reg.x.cx = 1;
130:    i_reg.x.dx = 1;
131:
132:    int86( 0x25, &i_reg, &o_reg );
133:
134:    sts = 0x000f & o_reg.h.al;
135:    if( sts==2 ) {
136:       sts = 0;
137:    }
138:    else {
139:       sts = 2;
140:    }
141:
142:    return( sts );
143: }
```

リスト5

```
 1: /***********************************************************/
 2: /*r_mediac メディア交換処理(RAMなので未使用のはず)      */
 3: /***********************************************************/
 4: int r_mediac( req_chg )
 5: struct REQ_CHG *req_chg;
 6: {
 7:   int   sts;
 8:   int   n;
 9:
10:         req_chg->diskfg = 0L;      /* stsコードセット        */
11:
12:         if( (sts=blk_out( &(req_chg->diskfg), sizeof(req_chg->
diskfg) ))) {
13:         /* リクエストヘッダ送信 */
14:                ;
15:         }
16:
17:         return( 0 );
18: }
```

▶苦労してスクリューアッパーを出しているのに画面の端で「ふん」だと。許せん，キム・カッファン。
高橋　博樹(23)京都府

第95回

猫とコンピュータ

マドだらけの話

Takazawa Kyoko
高沢 恭子

キョウコさんちに新しい仲間がやってきました。かねてよりウワサのドスブイです。初めてのマシン環境そしてWindowsからは，マンションの窓と同じように，いろいろな景色が見えてきます。

窓の下の楽園

三重県で過ごす日々の最大のオマケといえるのが，マンションの南側の家庭菜園でくりひろげられる，猫ファミリーの連載絵物語である。

手入れの行きとどいた緑ゆたかな菜園で昼寝や追いかけっこをしている猫たちは，いつも，誰からも叱られない。なぜかみんな仲よしでしつけのいい猫なのだ。

菜園にはトマトやナス，カボチャといったものももちろんあるが，バラやぼたん，ショウブ，ムクゲやタチアオイなどが，季節を追って花を咲かせる。野菜の緑と花との調和がなんともいえずみずみずしく，その植え込みをぬって，白，黒，茶トラ，シャムなど，10匹ほどの猫たちがはねまわっているようすは，まったく童話さながらのほほえましい光景だ。

マンションの周囲はすべてといっていいほどの，一戸建ての家並みである。北側はこのマンションのオーナーが経営するスイミングスクールとスーパーマーケットで，広い駐車場つき。南のベランダ側の一角に，マンションの敷地と接してほぼ正方形の200坪ほどの菜園がある。その位置はちょうど私たちの部屋の正面であり，2階という視点の高さも格好で，ほんとによい眺めなのである。

菜園の持ち主は船員だった人で，退職後に家庭菜園を楽しむために土地を購入したのだそうだ。少し広いので，そばに住む人たちにも土地を提供して，いっしょに趣味を楽しんでいるということである。野菜にかぎらずいろいろな種類の植物が栽培されているのは，何人かの人たちがそれぞれ好みのものを育てているからだろう。

猫たちが仲よしなのはみんな同じ家で飼われている一族だからだということがわかったのは最近のことだ。飼い主の住まいは菜園の一辺にあたるブロック塀に囲われた大きな洋風の邸だった。大の猫好きという奥さんは菜園のメンバーで，しつけのいいきれいな猫たちはここで一日中思いっきり遊びまわっても叱られるはずがないわけだ。

広い屋敷に飼われている猫たちは，日に何回か，そろってブロック塀の飾り穴からつぎつぎ菜園にあらわれてひとときを過ごし，また屋敷にもどっていく。白い子猫がトンボにおどろいて後ずさりする。相棒の茶トラの子猫が手を出してアタックしてみる。母親の白猫はそれを眺めながらのんびり体を横たえる。サトイモの葉陰には，2匹の黒猫が目をのぞかせている。

猫の集団といえば対立と抗争をくりかえすものばかりを見てきたのに，この町の，この部屋に暮らしたことで，ほほえましい猫一族のドラマを，毎日，リビングから楽しむことができる。これは，ちょっとした幸せといえる。

ドスブイはソフトです

私がDOS/Vを新機種の名前だと思っていたと聞いて，DOS/Vを専門にあつかうY氏は苦笑しながらいった。

「一太郎やロータスを使っている人で，そう思っている人は多いですよ」

このところのにわか勉強で，DOS/Vの入門書を数冊ひろい読みした効果があって，基本的なことは少々わかった。

話題となって登場したのは3年も前のことらしく，入門書も1992年ころ出版されたものが多かった。DOS/V時代の到来はこれまでも何回かウワサされながら，ほんとうの普及にはならず，今回は3度目くらいの波なのだそうだ。

日本ではPC-98シリーズが主流であり，日本のパソコンを育ててきたのもPC-9801パソコンといえる。PC-9801には「一太郎」や「Lotus1-2-3」などのヒットソフトもあり，PC-9801に対応して開発されたアプリケーションソフトは豊富である。

しかし世界に目を向けるとPC-98の市場はないにひとしい。世界市場でそのほとんどを占めているのが，IBM-PCとその互換機である。

どの本もこういうことを前提に話をはじめている。IBMが市場を大きくしたのは互換機の開発をはじめとしたメーカーの努力によるもので，市場が大きくなったことで，低価格ですぐれた性能のマシンを多くの人が使えるようになった。流通しているソフトも豊富である。

IBM社はIBM DOSの内容とともに，日本語処理のプログラムを作って公開し，共通のOSで動くパソコンを日本の多くのメーカーが作れるようにした。DOS/Vとは，基本のIBM DOS（英語版）に日本語処理のプログラムを加えたソフトの名称である。

DOS/Vマシンとしてのハードウェアの条件はどんなものか。

IBMパソコンかその互換機であること。画面解像度がVGA（640×480ドット）以上であること。CPUは80286以上。メモリ1Mバイト以上，ハードディスク10Mバイト以上であること。

これらは最低の条件で，この上で走らせるソフトによってはもっと上の性能が必要となる。Windowsなどを使うと，実用的なレベルでいえば，CPUは80486以上，メモリ6Mから8Mバイト以上，ハードディスクは200Mバイトは必要だ。

共通の規格であるDOS/Vを読みこませ

ることによって動くマシン。これを作るメーカーは三洋，東芝，日立，松下，三菱など，著名なものだけでも20社近くあり，それぞれがDOS/Vパソコンとして自社の特色を出しながら製作している。

またハードウェアの各部を個々に各社から選んで，個人で組み立ててマシンを構成することもできるので，ハードに精通しているマニアの人たちには新しい楽しみでもあるようだ。

いままでは自分のパソコンの環境をさらに高めようとするとき，すでに自分がそなえている機器のメーカーや専用のソフトにしばられてきたのものだ。

しかし，DOS/V仕様のパソコンならば，新しい要望が出てきたとき，自分のイメージする機能に合わせて，基板やボードの差し替え，追加などでいつでもグレードアップができる。しかも多くのメーカーから選べる。価格も安く高性能がポイント。当然ソフトも豊富。長い目で見たときもっとも経済的，建設的である。

窓のなかのソフト

Y氏のお世話で各部のハードウェアも取りそろい，盛夏のある日，わが家のDOS/Vパソコンが到着した。

CRTとハードが組み込まれたケースは体積だけでも貫禄じゅうぶんだ。

CPUは486DX2，66MHz，ビデオボードは1280×1020，内蔵メモリは16Mバイト，ハードディスクは540Mバイト。これに音源ボードを加えた。

DOS/V専用の17インチのディスプレイ，キーボードとマウス。

MS-DOS6.2/VにはWindows3.1もつけた。ドスブイといえばWindowsマシンともいうべきで，大量のハードディスクはWindowsのためといってもいい。

Windowsは誰もが認める「ハードウェア資源の大食漢」なのだそうだ。しかしこれをベースにたくさんのソフトを同時に画面上で進行させる，マルチタスクの実現のためにDOS/Vパソコンはあるようなものなのだ。

ともかくそのウワサのWindowsを開いて，「お絵描き」ソフトあたりをのぞいてみることにする。

電源を入れると40秒ほど内部チェックがあり，そのあとドライブがハードディスクにいく。「WIN」＋リターンキーでWindowsになる。

アイコンの並ぶ画面をつぎつぎクリックして，窓を開けたり閉じたりしてみる。ウィンドウやアイコンならX68000でおなじみと，よく知っているつもりでマウスを動かしてみるけれど，どうも考えていたより複雑でややこしい。

誰にでもわかる絵表示だなんてウタイ文句にのせられて，マニュアルは読まなくてもいいと思ったのが甘かった。知らないうちにどこかの部分をクリックしてしまって，窓も画面も行方不明になる。さあタイヘンとあっちこっちさがすうちに，別の窓に触れてちがう操作をしたらしい。

「X68000はこんなじゃないのに」と，さらにメチャメチャにクリックする私に，「シングルタスクとマルチタスクはちがうからね」と夫がそばからいう。

パソコンを使うために覚えなくてはならないことと，身につけることを，最小限にしたといわれるWindowsでも，操作になじむための練習は必要だ。いくらなんでも，いきなりの突撃はまずかった。

それでもなんとか「アクセサリ」のアイコンから，「ペイントブラシ」のシンボルマークをみつけて窓を開いた。

X68000の「Z'sSTAFF」にくらべたら，小学生のおエカキくらいの色数で，ペンの種類もすくないけれど，けっこういろいろなことが楽しめる。早くそれを印刷して，ドットの繊細さをたしかめたいものだ。

ところがプリンタへの出力で挫折した。なんとしても印刷できない。プリンタの設定を自己診断するソフトもあって，こまかい項目のチェックもできるが，それだけでは原因がつきとめられない。Y氏が急きょ運んでくれたエプソンのレーザープリンタでトライしてみたらなぜか印刷できたのだが，わが家のプリンタで印刷できないのでは困る。いま究明中である。

illustration：Kyoko Takazawa

折も折，狛江のアニキから電話があった。

「DOS/V入れたんだって？　何するつもりだい？」

「……マア，絵を描いたり，文章書いたり……私だけが使うわけじゃないし……」

なんだか返答にならない。

「じつはボクもね，DOS/Vの講習会に出かけるつもりで申しこんでおいたんだけどね，都合が悪くて行けなかったんだよ」

よく聞けば，それもキッカケになり，やっぱりドスブイの主義には従わずに，PC-98路線で進むつもりらしい。

自分のマシンやシステムは，自分の主義や主張がだんだんと積み重なっていくものでもある。たやすく新しいスタイルに乗り換えるにわけにはいかない。

DOS/VやWindowsの構想はいままでとはまったくちがう世界だ。だから入っていくには抵抗もある。アニキの気持ちはよくわかる。それでも，「いったい，何やるつもりだい？」という質問への回答がそのままこれからの楽しみでもある。

ANOTHER
CG
WORLD
VOL. 38
©KYOKO EGUCHI

1

2

3

ふむ
これから友だちと
おいしいもの作って毎日
パーティーするんだってェ
おさそい
うれしい
けど…

あたしゃ
食欲の秋じゃなくて
今年はX68000で
知性の秋にする
のよー
ボンノウ
こらえて

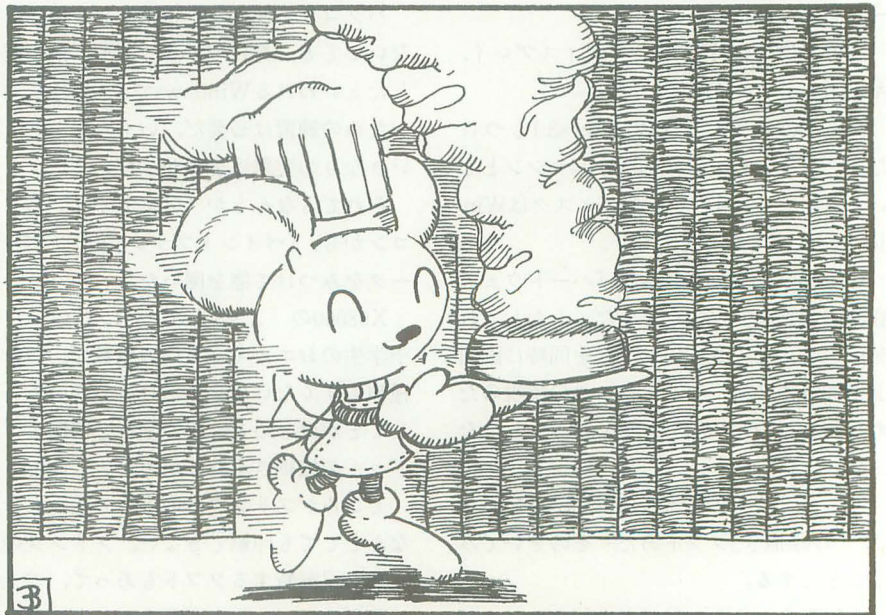
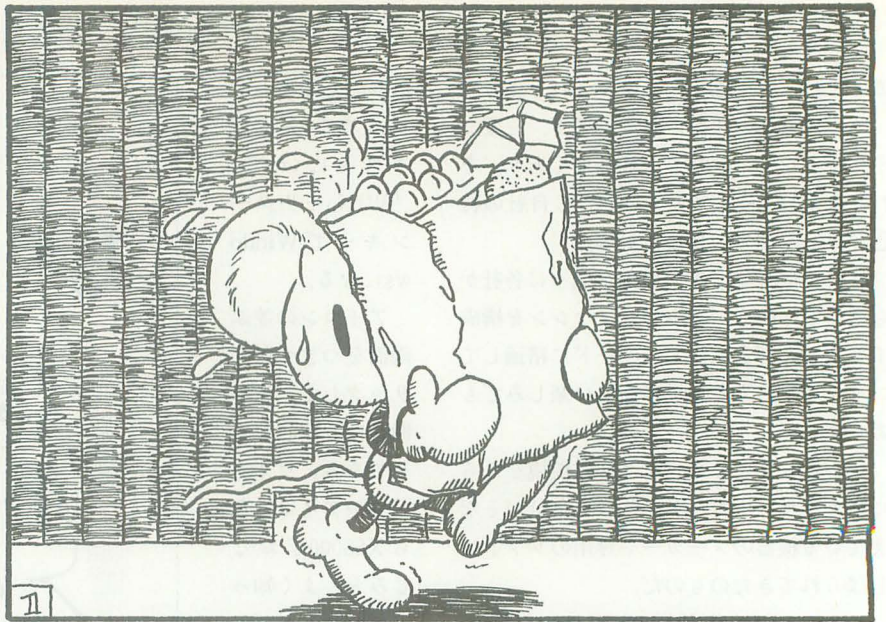
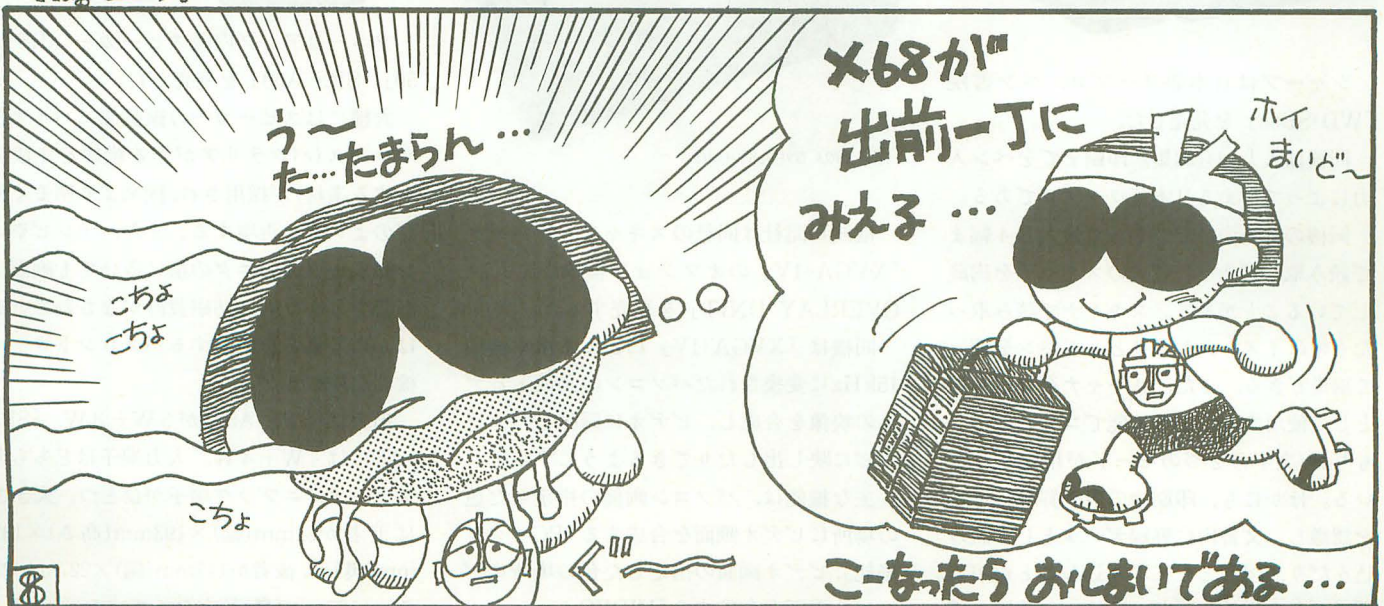
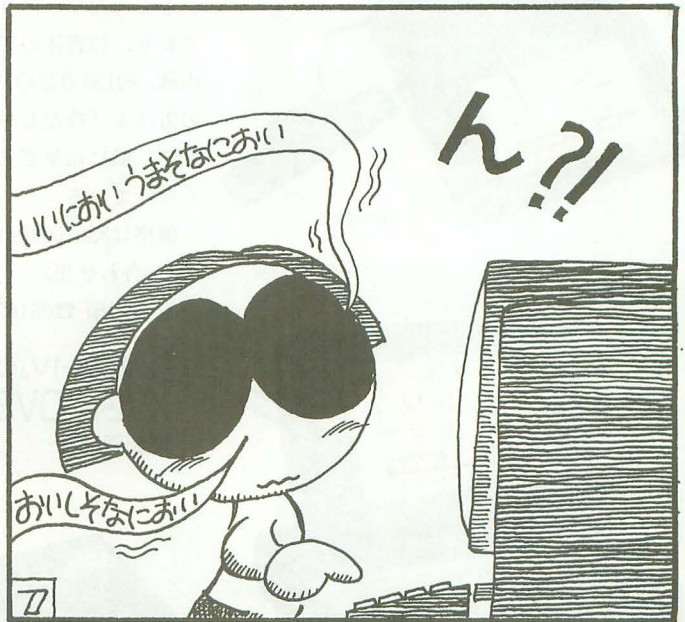
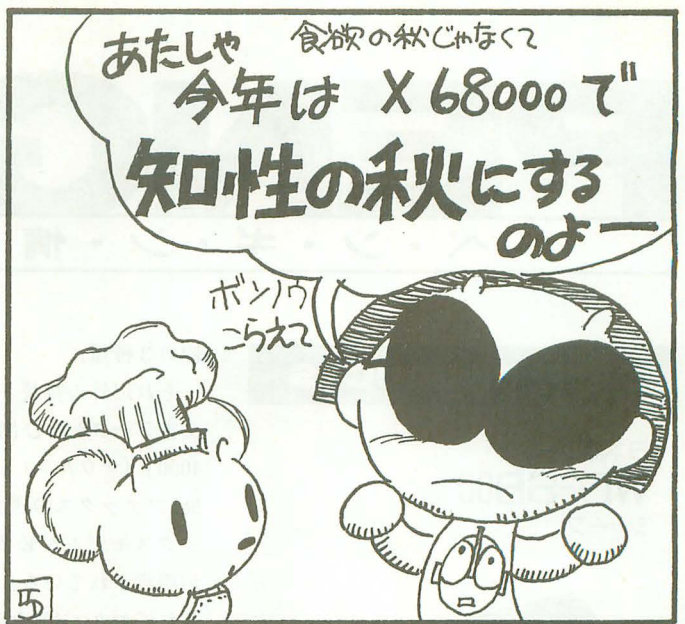
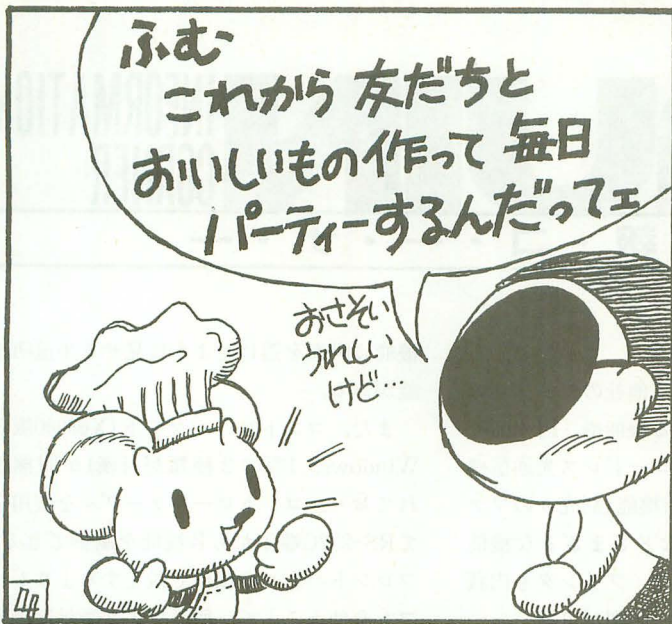
澄んだ空気の中
人間とマシンとの静かなる対話を
楽しむ…… 秋…

ん?!!
いいにおいうまそなにおい
おいしそなにおい

Aug 21 '94

う～
た…たまらん…
こちょ
こちょ
こちょ
ぐ～～
X68が
出前一丁に
みえる…
ホイ
まいど～
こーなったら おしまいである…

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

日本語ワープロ
WD-SB60
シャープ

WD-SB60

シャープは日本語ワープロ，ペン書院「WD-SB60」を発売した。

同機は入力から編集，印刷までをペン入力によって行える日本語ワープロである。

同機の特長のひとつとして最大B４幅まで読み取り可能な着脱式のスキャナを内蔵していることがある。スキャナで読み取ったものはイメージデータとしてペンを使って編集できる。また，スキャナをコピー機として使用することも可能でコピーモードも写真/文字の２つのモードが用意されている。ほかにも，印刷物の表や罫線の情報を認識し，文書内に罫線データとして読み込んだり，図形として取り込むことが可能。認識できる罫線の種類は実線，二重線，破線の３種類。

それ以外の特長として他社のワープロの文書データが呼び出せる機能や「PI-3000/4000」(ザウルス)とのコードレス光通信機能，ファックスの送受信機能(別売りのファックスモデムが必要)などさまざまな機能が用意されている。また，プリンタも内蔵されており，400dpiでの印刷が可能。

もちろんワープロとしての機能も充実しており，13書体のアウトラインフォントを内蔵，約130万語の辞書を搭載し，ペン入力の方法も「枠なしペン入力」を採用することで，紙に鉛筆で文字を書く感覚により近くなっている。

価格は330,000円(税別)。

〈問い合わせ先〉
シャープ㈱ ☎06(621)1221，03(5261)7271

「XVGA-1V」のオプション機器
XVGA OVERLAY UNIT
電波新聞社

XVGA OVERLAY UNIT

電波新聞社は同社のスキャンコンバータ「XVGA-1V」のオプション機器「XVGA OVERLAY UNIT」を発売する。

同機は「XVGA-1V」によって水平同期15kHzに変換されたパソコンの画面とビデオの映像を合成し，ビデオに録画したりテレビに映し出したりできるようにする。

主な機能は，パソコン画面の指定した色の場所にビデオ画面を合成する「PCマクロ機能」，ビデオ画面の指定した色の場所にパソコン画面を合成する「VIDEOクロマキー機能」，画面を透けたように見せる半透明機能がある。

また，コントロールソフト(X68000版とWindows3.1版の２種類が付属)が同梱されており，コントロールケーブルを使用してRS-232C端子から各機能を制御できる。フロントパネルの操作で設定するよりもソフトを使うことで，簡単でより微妙な調整ができるようになっている。

価格は45,800円(税別)。

〈問い合わせ先〉
マイコンソフト㈱ ☎06(203)2827

アンプ内蔵スピーカー
SRS-A50/SRS-A60
ソニー

SRS-A60
ソニーはアンプ内蔵スピーカー「SRS-A50」「SRS-A60」を発売した。

両機にはスピーカーの振動部にバイオセルロース(バクテリアが作る繊維を主成分とする素材)が採用され，従来より明るく伸びのよい音を再現する。また，テレビやコンピュータのモニタの横に置いても画面に影響を与えにくい防磁設計がなされている。ほかにも低音を調整するバスコントロール機能を搭載した。

出力は「SRS-A50」が３W＋３W，「SRS-A60」は４W＋４W。入力端子はどちらもステレオミニプラグ端子がひとつ。大きさは前者が82mm(幅)×193mm(高さ)×132mm(奥行)，後者が117mm(幅)×227mm(高さ)×130mm(奥行)となっている。

価格は「SRS-A50」が8,800円,「SRS-A60」が11,000円(ともに税別)。
〈問い合わせ先〉
ソニー㈱　☎03(5448)3311, 06(251)5111

デジタルスチルレコーダ
SR-3100
オリンパス光学

SR-3100

オリンパス光学はデジタルスチルレコーダ「SR-3100」を発売した。

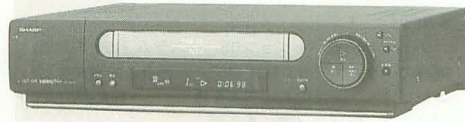
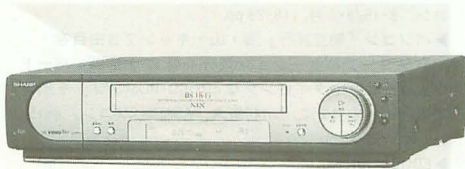
同機は外部モデムを介して一般公衆回線に接続して画像電送が行える。送受信先としては,「SR-3100」,同社のデジタルスチルカメラ「VC-1100」,PC-9801シリーズ(同社の画像電送ソフトを利用),NIFTY-Serveのメールボックスなどが選択可能。また,10カ所までの登録ができる自動ダイヤル機能や呼び出しに自動的に応答するリクエスト機能,タイマー送受信機能などが用意されている。

扱う映像に関してはS映像信号/RGB映像信号に対応しており,高解像度ビデオカメラの接続によりNTSC最高水準の静止画の記録が可能。記録媒体にはPCカードと3.5インチフロッピーディスクを採用し,記録方法はJEIDA対応,画像圧縮規格はJPEG方式に対応している。

価格は650,000円(税別)。
〈問い合わせ先〉
オリンパス光学㈱　☎03(3251)8028

VHSビデオカセットレコーダ
VC-HF35/VC-BF55
シャープ

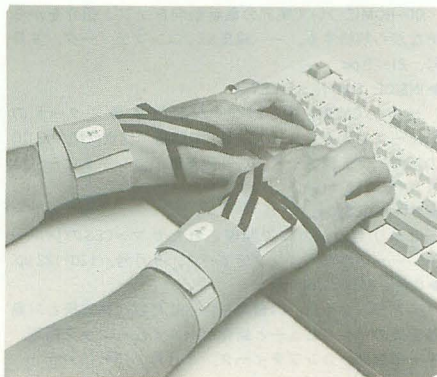
シャープはVHSビデオカセットレコーダ2機種「VC-HF35」「VC-BF55」を発売した。

「VC-HF35」はビデオソフトなどの英語字幕を再生できる「クローズドキャプション」に対応している。字幕表示の切り替えはリモコンによって行う。ほかにも,スポーツ中継の時間延長に自動対応する「シネ・スポ延長タイマー機能」により録画ミスの可能性が減った。また,新聞や雑誌に掲載される番組ごとのGコードを入力すれば録画予約ができる「Gコード予約」システムをリモコンに内蔵している。

VC-HF35

VC-BF55

「VC-BF55」は「VC-HF35」の機能に加え,BSチューナーを内蔵している。ほかには,データ表示部に2色発光の液晶ディスプレイを採用し,表示色の違いで現在の動作状況を確認できるようになっている。

価格は「VC-HF35」が66,000円,「VC-BF55」が77,000円(ともに税別)。
〈問い合わせ先〉
シャープ㈱　☎06(621)1221, 03(5261)7271

腱鞘炎を予防する
リスト・サポート
ウエルネット

リスト・サポート

ウエルネットはキーボードを長時間使用する人がかかりやすい腱鞘炎を予防する「リスト・サポート」を発売した。

同商品は指を自由に動かせる状態のまま手首だけを固定するので,安定感があり手首への負担が少ない。裏地には特殊起毛素材が使用され,装着部分に熱や湿気がこもらないようになっている。また,洗濯機で洗って,いつまでも清潔に使うことができる。色は黒のみ。重さは57g(片手)。

価格は8,000円(左右1組,税別)。
〈問い合わせ先〉
㈱ウエルネット　☎03(3562)3688

INFORMATION

エレクトロニクスショー'94
フォーラム開設
日本エレクトロニクスショー協会

日本エレクトロニクスショー協会は10月4日～8日に東京,晴海で開催する「エレクトロニクスショー'94」の情報をパソコン通信で公開する。

フォーラム開設期間は9月1日～10月31日。開設場所はPC-VANが「ショッピング・情報ファイル」内,NIFTY-Serveは「フォーラム」内になる。ダイレクトアクセスは順に「JELESHOW」と「GO FELESHOW」で行う。

フォーラムは情報発信エリア,コミュニケーションエリア,データライブラリーの3つで構成されている。その中から,開催概要,出展各社の展示情報,セミナー情報,ニュースなどを知ることができる。
〈問い合わせ先〉
日本エレクトロニクスショー協会☎03(3284)1051

ソフトバンクブック
全点フェア
旭屋書店札幌店

旭屋書店札幌店では,ソフトバンクより発売された最新刊から手に入れにくい本まですべてを集めた「ソフトバンクブック全点フェア」を開催する。また,ソフトバンクが発行している雑誌のバックナンバーも同時に販売している。

期間は9月30日まで。場所は旭屋書店札幌店(地下鉄すすきの駅,大通り駅より徒歩5分)。
〈問い合わせ先〉
旭屋書店札幌店　☎011(241)3007
〒060 札幌市中央区南三条西4丁目
アルシュビルB1・B2

FILES

Oh!X

このインデックスは，タイトル，注記――著者名，誌名，月号，ページで構成されています。虫の鳴き声が聞こえてくる今日この頃。日が暮れるのも早くなりました。スポーツ，読書，芸術など皆さんそれぞれの秋を満喫してください。

参考文献
I/O　工学社
ASAHIパソコン　朝日新聞社
ASCII　アスキー
コンプティーク　角川書店
C MAGAZINE　ソフトバンク
電撃王　主婦の友社
マイコンBASIC Magazine　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

一般

▶NEWS
Macintoshの新バージョンOS「システム7.5」の話題，東芝の「ダイナブック限定発売」企画中止の事件など，コンピュータ関連のニュースを掲載。――編集部，ASAHIパソコン，8・15/9・1号，8-9 pp.

▶ナイスバディを探せ！
7機種のサブノートパソコンをピックアップして重量やインタフェイスなどの性能を比較する。「ThinkPad 230Cs」と「DynaBookSS」が高評価。――編集部，ASAHIパソコン，8・15/9・1号，18-29 pp.

▶パソコン「架空旅行」海・山・キャンプ自由自在
パソコンでバケーション気分を味わう企画。バードウォッチングやダイビングなどのCD-ROMソフトを紹介する。――中井紀之・川村敏久，ASAHIパソコン，8・15/9・1号，30-38 pp.

▶EDUCATION
パソコン教育に関するニュースのページ。大手の事務機総合商社，内田洋行が小学校向けの性教育ソフトの販売を開始。――坂本伸之，ASAHIパソコン，8・15/9・1号，48-49 pp.

▶パリスの審判
架空ソフト発明のコーナー，今回の作品は「オキシジェン・デストロイヤー」。二酸化炭素排出量をめぐって駆け引きを繰り広げるシミュレーション。ほかには泳法マスターシミュレーションなど。――井上正基，ASAHIパソコン，8・15/9・1号，128-130 pp.

▶機械用言博物館　15
コンピュータのスピードを表す言葉について考える。「速い」「軽い」など。――荻窪圭，ASAHIパソコン，8・15/9・1号，148-149 pp.

▶ハードウェアFLASH!
ハードディスク「VESPER」，ページプリンタ「Pana Printer」などハードウェアの新製品情報。――編集部，LOGIN，16・17合併号，40-45 pp.

▶THE NEWS FILE
セガの「ジョイポリス」とナムコの「たまご帝国」オープンの話題などハイテク関連のニュース。――編集部，LOGIN，16・17合併号，46-51 pp.

▶おそるべしお子様電子手帳
各社から発売されている子供用電子手帳を特集。――編集部，LOGIN，16・17合併号，204-207 pp.

▶IBM-PC入力デバイス品評会
IBM-PCの入力デバイス「FORMULA T1」「CONTROL SYSTEMシリーズ」などを紹介する。――編集部，LOGIN，16・17合併号，210-213 pp.

▶架想楽園へ行こう Ver.2.03　No.7
VRテクノロジー満載のテーマパーク「たまご帝国」を取材する。――中田宏之，LOGIN，16・17合併号，226-229pp.

▶くねくね科学探検　第4回
今回は隕石が地球に衝突する可能性について考えていく。――鹿野司，LOGIN，16・17合併号，230-233 pp.

▶特集　広がるCD-ROMソフトの世界
CD-ROMについて業界の最新動向をソフト紹介をからめながら解析する。――編集部，コンプティーク，9月号，21-29 pp.

▶INSIDE THE NEXT LEVEL
セガのCATVを使ったゲーム配信事業スタートのニュースや「インターネット」の話題，次世代機や3DOの新作ソフト情報など。――編集部，コンプティーク，9月号，114-115 pp.

▶サマーCES報告
アメリカのシカゴで開催されたサマーCESのレポート。――編集部，コンプティーク，9月号，120-122 pp.

▶Game Music Journal
「ベアナックルIII」の音楽を担当した古代祐三氏と川島基宏氏のインタビューと新作ゲームミュージック情報。――編集部，コンプティーク，9月号，134-135 pp.

▶NEWS COLLECTORS
任天堂の32ビットマシン「VU」や「PlayStation」のソフト流通，セガとナムコがオープンしたアミューズメントスペースの情報など。――編集部，電撃王，9月号，22-25 pp.

▶激突への秒読み
「PlayStation」の第2回進捗報告会の模様，「SEGA SATURN」の「バーチャファイター」開発情報，「PC-FX」についてのハドソンへのインタビューなど。――編集部，電撃王，9月号，28-39 pp.

▶不滅の100タイトル
次世代機戦争のカギを握ると思われる現行機種のヒット作100本をセレクト。各次世代機への移植可能性を判定する。――編集部，電撃王，9月号，40-53 pp.

▶DENGEKI SUPER HIT CHART
コンシューマソフトの販売動向や，パソコンゲームの売り上げTOP 10などの情報を掲載。――編集部，電撃王，9月号，92-99 pp.

▶特集　フルカラー・プリント環境を手に入れよう！
カラープリンタやビデオプリンタ数機種を取り上げ，それぞれの特徴を解説。――編集部，マイコンBASIC Magazine，9月号，35-45 pp.

▶CD-ROMからはじめるマルチメディア　第4回
「ビデオCD」を取り上げ，再生するハードをチェックしていく。――吉岡哲也，マイコンBASIC Magazine，9月号，52-55 pp.

▶新ハード特捜部
「PlayStation」の第2回進捗報告会のレポートと「SEGA SATURN」「スーパー32X」の開発中ソフトの画面を一挙掲載。――編集部，マイコンBASIC Magazine，9月号，147-153 pp.

▶Arcade Game Graffiti　第7回
1980年の最終回。「ミサイルコマンド」などのATARI社のゲームを特集する。1980年の発売ゲームリストと㈱キョーワの鈴木社長へのインタビューつき。――編集部，マイコンBASIC Magazine，9月号，154-157 pp.

▶特集　アプリケーション3種の神器
ワープロ，表計算，データベースの3分野を代表するソフトウェアの操作性，機能などを検証する。――編集部，I/O，9月号，39-57 pp.

▶最新技術情報
「PCIローカルバス」と「ビデオ・オン・デマンド技術」について技術的な側面を解説する。――田嶋孝行，高間剛典，I/O，9月号，69-76 pp.

▶NEWS BYTES
「PowerPC 603」の出荷状況などパソコン市場の世界的なニュースを取り上げる。――編集部，I/O，9月号，81-86 pp.

▶特集　パソコン・レスキュー隊
マシンごとのトラブル救済方法から，OSなどパソコンに関する疑問を徹底解決。――編集部，I/O，9月号，97-108 pp.

▶MultiMedia Watching　第9回
FM文字多重放送を使った「見えるラジオ」などマルチメディアをうたう新製品とそれにまつわる事情を紹介。――奥野雅之，I/O，9月号，134-137 pp.

▶特集2　インターネットのこと全部教えます
インターネットへのアクセスから実際の利用まで，インターネットの基本から応用までを解説。――編集部，ASCII，9月号，245-268，496-499 pp.

▶魅惑のニューテクノロジー　第6回
動画像圧縮のフォーマットとして定着しつつあるMPEG方式の圧縮技術とその応用を紹介。――編集部，ASCII，9月号，318-323 pp.

▶INTERCOOLED
「PlayStation」のソフト開発状況や「3DO」の新作情報など。――編集部，ASCII，9月号，324-327 pp.

▶スペシャルインタビュー　第17回
元海軍技術研究所に勤務されていた山本正治氏に戦時下での日本の情報通信技術事情を聞く。――編集部，ASCII，9月号，355-359 pp.

▶DIGI-VIS TODAY
IBMが生んだ超高速グラフィックエンジン「PVS」を紹介する。――聖咲奇，ASCII，9月号，382-383 pp.

▶稀代もののけ考
おきらくハイテックグッズとしてコンパクトカメラや電子辞書，録音ペンなどを紹介する。――編集部，ASCII，9月号，384-385 pp.

▶電子頭脳映画史　22.教訓：デロスの悲劇
「ウエストワールド」と「未来世界」を取り上げ，現代のテクノロジーの進化を考察する。――聖咲奇，ASCII，9月号，386 p.
▶アスキーロードテスト No.58
ザウルスの試用レポートの4回目。フラッシュメモリの使い方を考える。ほかに新ザウルスのFAXモデムを試してみる。――宮野友彦，ASCII，9月号，452-453 pp.
▶特集　ノート・サブノートパソコンの選択と活用
ノートパソコン6機種の実用性を比較検証する。――編集部，My Computer Magazine，9月号，9-28 pp.
▶光磁気ディスクの勧め　その2
光磁気ディスクの購入からセッティングまでを筆者の体験をもとにアドバイスする。――佐田守弘，My Computer Magazine，9月号，45-49 pp.
▶レッツ・プログラム
分数の覆面算をテーマに読者から寄せられたプログラムを分析する。――藤本健，My Computer Magazine，9月号，66-72 pp.
▶パソコン研究室
MOドライブの仕組みと使い方について研究する。――Space Club，My Computer Magazine，9月号，90-93 pp.

MZシリーズ

MZ-2500(BASIC-M 25)
▶PACLIANT
地球を侵略しようとしている敵を倒すアクションゲーム。――KPC代表中原進，マイコンBASIC Magazine，9月号，90-92 pp.

X1 turbo/Z

X1シリーズ
▶みつぐ君
倉庫番風のパズルゲーム。――岡崎隆，マイコンBASIC Magazine，9月号，109-111 pp.

X68000

▶NEWSOFT
X68000用「スーパーストリートファイターII」の情報を掲載。――編集部，LOGIN，16・17合併号，20 p.
▶GameReview
「餓狼伝説スペシャル」を取り上げる。――Ryu.宮澤，LOGIN，16・17合併号，297 p.
▶電撃王全ゲームインデックス
9月に発売予定のゲームリスト。9月号に載っているソフトのインデックスつき。X68000用は「餓狼伝説SPECIAL」など。――編集部，電撃王，9月号，8-10 pp.
▶新作王
X68000用では「スーパーストリートファイターII」移植決定のニュースなど。――編集部，電撃王，9月号，165 p.
▶S.S.Express
新作ソフトの情報コーナー。特集では同人ソフトを取り上げる。X68000用の「FIFTEEN ALL」や「ARTEMIS」が登場。ほかには「スーパーストリートファイターII」移植のニュースなど。――編集部，コンプティーク，9月号，40，41，54 pp.
▶機種別SUPER SOFT INDEX
発売予定のソフトを機種別，発売日順に掲載。X68000用は「魔法大作戦」など。――編集部，コンプティーク，9月号，101 p.
▶TREASURE CAVE
ロープを使って宝石を取るアクションゲーム。――パト，マイコンBASIC Magazine，9月号，112-114 pp.
▶Crystal Zone
脱出タイプのパズルゲーム。――橘宏之，マイコンBASIC Magazine，9月号，115-118 pp.
▶ナックルヘッズ～鷹舞狼吼～
ナムコの「ナックルヘッズ」のミュージックプログラム。――上古仁志，マイコンBASIC Magazine，9月号，127-129 pp.
▶AV STRASSE
「040 turbo」を紹介。仕組みからソフトウェアの変化，ベンチマークテストなど。――編集部，ASCII，9月号，342-344 pp.
▶ONLINE SOFTWARE INDEX
大手ネットにアップロードされたソフトを紹介する。X68000用は「fdx.x」「CDROM.X」のファイルツール2本。――編集部，ASCII，9月号，431 p.
▶なんでもQ＆A
「SX-WINDOW ver 3.1」のコンソールモードとシャーペンのイメージペーストによる印刷の疑問に答える。――シャープ，My Computer Magazine，9月号，148-149 pp.
▶SX-WINDOWプログラミング　第11回
XGCCでシャーペンの外部コマンドを作成するためのライブラリを作成する。――吉野智興，C MAGAZINE，9・10月合併号，110-115 pp.

ポケコン

PC-E500
▶土龍の逆襲
もぐらたたきゲーム。もぐらがフェイントをかけてくるのがミソ。――西野陽一，マイコンBASIC Magazine，9月号，119-120 pp.

新刊書案内

インターネットの世界
月刊マックライフ編
ビー・エヌ・エヌ刊
☎03(3238)1622
四六判　175ページ
1,600円（税込）

いまさらだが，インターネット流行で，どこから出てきたのか，インターネット関係の本が10冊以上平積みになっていたりする。凄いことだ。なかには研究者が書いたUNIXベースのもの，技術的側面やインターネットの構造をベースに解説している普通の人にはちょっと難しいもの，翻訳書でよくできているが日本の個人ユーザーが読むには難解だと思われるものが多い。そのなかから，インターネットの概念やその意義，具体的な接続までを個人ユーザーでも読んで楽しいようにまとめられているものを探したら，本書に辿り着いた。月刊MACLIFE編集部編であり，内容がMacintoshに偏ることもないことはないが，読んでいて気にならないレベルだ。Macintoshって文字が頻出することもなく編集部による体験記以外は非Macintoshユーザーでも全然問題はない。
本書は複数の人間による解説（この手の話題ではお馴染みの浜野保樹，『SPA!』でも連載している橋本典明など）を採用し，かなり読みやすい。それぞれが勝手に好きなことをいうのではなく，浜野氏がわかりやすくインターネットの成り立ちや意義について歴史的に解説し（パソコンユーザーには非常にわかりやすい），日本でインターネットのプロジェクトを推進しているjunetを設立した村井氏が日本で行っているWIDEプロジェクトを紹介している（すでに全国的にISDN接続されているコンビニがインターネットのプロバイダーになれば面白いという説は凄い）。ほかにも国際GLOCOM研究員新谷氏がインターネットのアカデミック，UNIX，TCP/IPという3つの呪縛からの解放について話をし，橋本氏が感覚的にインターネットの世界を述べて，マイナーなものの集積こそがマルチメディアの世界だと語る。これだけで，かなり見えてくる。ここでやっと技術的なインターネットの概念や具体的な接続方法の話へと至っていく。とりあえず読むにはお勧め。　（K）

ポイント図解式
最新MPEG教科書
藤原　洋監修
マルチメディア通信研究会編
アスキー刊
☎03(5351)8194
B5判　292ページ
2,900円（税込）

マルチメディアが声高に叫ばれる昨今，「MPEG」という言葉もよく耳にするだろう。この言葉は「Moving Picture Experts Group」の略でメディア統合系動画像圧縮の国際標準を指す。
本書は，JPEGやMPEG Iなどのマルチメディアに関する規格や方式を，図をふんだんに使ってわかりやすく解説している。それ以外にも，情報圧縮の基礎的な知識やマルチメディアパソコンに求められる性能について解説したり，各規格の標準化への歴史が紹介されている。
専門家や技術者だけでなく，マルチメディア技術に興味のある方にお勧めである。

日本人と数
江戸庶民の数学
佐藤健一著
東洋書店刊
☎03(3269)2961
四六判　235ページ
1,800円（税込）

江戸時代に日本全国では約2万軒以上の寺子屋が存在したといわれている。そこでは「読み・書き・そろばん」が教えられ，数学の知識が多くの庶民に普及した。もちろんその知識が生活していくうえで必要だったからである。
本書ではそんな江戸時代の日常生活のなかで使われた数学について触れている。たとえば，商人が必要とした数学，ねずみ算やまま子だてなどの遊びの数学，神社や仏閣に数学の問題とその答えを奉納した「算学奉納」など。
勉強としての数学ではなく，文化としての数学に触れてみるのもおもしろいかもしれない。

瀧さんのメガディスプレイ計画で，15kHzモードを捨てろと書いてありますが，私はどうしても15kHzモードを捨てることができません。ソフト的にではなく，ハード的になんとか15kHzモードを30kHz以上にすることはできないでしょうか？　こうすれば，ビデオなどもマルチスキャンモニタに映すことができます。

私自身も，このようなことをする機器がないかと調べて見たところ，ナナオからErgoVerter98というものが発売されていました。これはPC-9801の24kHzモードを31kHzモードで映すというもののようです。これと同じように15kHzから31kHzにする機械を，ローテクでなんとか作れないのでしょうか？

対応周波数が15kHz～80kHzぐらいまである，マルチスキャンモニタがあればよいのですが，どうやら探してもないみたいなので。　埼玉県　山本健二ほか同意見多数

名指しできているので，私が直接答えましょう。

この質問は何通りの意味にも取れるので，その都度考えられるパターンで答えます。

まず15kHzにこだわる理由ですが，単に映ればよいのでしょうか？　それとも，15kHzの映りが好きなのでしょうか？

実は私自身，画面を広くワイドに楽しめることと，1ドットが角張らないことから，マルチスキャンモニタを買う前は，ゲームは必ず15kHzモードでやっていました。同じような理由で15kHzモードが好きだった人なら，迷わずマルチスキャンモニタを買ってもよいと思います。画面をワイドにすることは，最近のマルチスキャンモニタはいくらでもすることができますし，その気になれば，オーバースキャン（X68000の15kHzモードのように上下左右が外にはみ出して映る）っぽく，広げることもできます。あまり古いやつでなければ，周波数帯ごとにそれらの画面情報をメモリに記憶することができます。

また，1ドットが角張るのが嫌なのならば，先月号のHighreso.xを多少改造すれば，31kHz以上でも1ドットが角張りません。X68000の256×256モードなどを高解像モードで映すと1ドットが角張ってくるのは1ラインを2度読みしているからで，これをやめれば（垂直同期周波数が一気に110Hzまで上がりますが），1ドットが四角くならなくなります。

当然その分，垂直同期周波数が上がるのでソフトによっては処理が速くなるおそれがありますが。このあたりは説明すると長くなるので，2回にわたる私の記事を読んでなぜそうなるかを考えてみましょう。

ハード的にこれらを行う方法としては，39MHzのオシレータを80MHzぐらいに取り替えるという方法があります。ただこれも垂直同期周波数が上がってしまうので注意が必要です。

15kHzにこだわる理由のなかに，NEO・GEOやメガドライブを映したいからという理由もあるでしょう。これらをなんとかするには，やはりスキャンコンバータなるものを使わなくてはいけません。

ErgoVerter98というのは，PC-9801の24kHz/53Hzの画面を31.5kHz/70HzのVGA仕様のスキャンモードにするもので，文字どおりスキャンコンバートをしています。

秘かに私はErgoVerter98を持っているのですが（ほとんど使っていない），これをX68000につけても24kHzモードがうまく31.5kHzモードになったりはしません。その理由はX68000のドットクロックにあります。98シリーズのドットクロックは，PC-H98，PC-9821シリーズを除いて，640×400モードでは一定です。したがって，このドットクロックを見込んでRGB各信号線に出てくるアナログ信号をサンプリングし，メモリーして，まったく別のビデオICから31.5kHz/70Hzで出力します。いわばこれはPC-9801専用であり，出てくる信号が完全に予測できるからできる技なのです。

X68000の15kHzのドットクロックは，固定ではありません。これらは，2回にわたるメガディスプレイ計画の原稿を見ていただければすぐわかると思います。256×256モードは39MHzのオシレータから作っていますが，512×256モードは69MHzから作っていたりします。X68000専用ならばイメージ端子から出力されるドットクロックを利用して作れないこともないのですが，専用にするならばソフトで回避しても同じです。

また，NEO・GEOやメガドライブ，PCエンジンなどの画面を30kHzモードで映したいという気持ちも山々ですが，これらのスキャンコンバートを「画質を劣化させずに」行うためには，専用にせざるをえないのです。

少々高くなってよいのならば，ローテクで，15kHzを31.5kHzにしてしまう汎用ハードはできないことはありません。ただ，そうなると画質の劣化は必至です。その理由は，水平の画像を表す水平同期期間が15.75kHz時より31.5kHz時のほうが短いからです。短いということは，アナログ信号を時間的に圧縮させることになるわけですから，劣化の理由はいうまでもありませんよね。

これをできるだけ劣化させずに再現するにはおそらく，ローテクでも10万以上かかってしまいそうな気がします。わざわざ31kHzにしたい理由の大半は，きっと画像が綺麗になるのではないか？　という憶測からきているのでしょう。それだと，この方法は却下ですよね。同様なことを行う高画質映像機器としてラインダブラーというものがありますが，だいたい100万円以上します。

15kHzにこだわる理由はいろいろありますが，画質を綺麗に，しかも安くと，そのように天秤にかけるならば，いちばんよいのは15kHzの映せるモニタを別に残しておくなり購入するなりすることではないでしょうか？

最後に，15kHz～31kHzが映せるマルチスキャンモニタが確か三菱から出ていますよ。かなり高そうですが，モニタが本体よりも高いのは当たり前ですからねぇ（編注：やはり普通はそんなことはない）。実質的寿命は本体よりも長いものですし。

（瀧　康史）

SX-WINDOW ver.3.1のコンソールウィンドウをデフォルトの大きさより小さく開きたいのですが，どのように指定したらいいのか教えてください。　神奈川県　大井　祐樹

質問とは関係ありませんが，大井さんの年齢を見て一瞬我が目を疑いました。10歳……。

では回答です。

まず問題になるのはどこを書き換えればいいのかということでしょう。コンソール自体の設定は，コンソール.ENVに入っています。試しにコンソール.ENVのスター

トアップマクロS0で大きさ指定をしてみました。するとコンソールの大きさは小さくできるのですが，ウィンドウサイズは元のままで妙に余白の多いウィンドウが開いてしまいます。コンソール機能の呼び出しはシャーペンやシステムアイコンから行われますから，最初に呼び出し側の設定を変更することにしましょう。

シャーペンを起動して環境アイコンのところでメニューを開き，キー定義の書き出しを選択します。次にいま書き出したファイルをシャーペンに読み込み，MENU 1（ディスクアイコン）の設定部分から，

'コンソール',#215,#70,'-N-1 -B$0f00 -Y$0100 -Gコンソール.ENV',#216,$0D

のような部分を探してください。

ここがコンソールを呼び出しているわけです。直接CONW.EX（コンソールの本体）を呼び出すのではなく，新規文書作成時にコンソール.ENVを指定することでコンソールを呼び出しているのがわかります。

ウィンドウの大きさはシャーペンの起動オプションで指定できます。マニュアルを見ると，

－Vn,n
　ウィンドウの幅と高さをドット単位で指定

というものがありますので，これを使いましょう。

先ほどの設定を，

'コンソール',#215,#70,'-N-1 -B$0f00 -Y$0100 -V400,320 -Gコンソール.ENV',#216,$0D

のように変更してみてください（大きさは好みで変更すること）。これを一度セーブしてシャーペンの「キー定義の読み込み」で読み直し，ついでに「キー定義と環境を保存」を実行します。

開いているコンソールがあれば閉じておいて，メニューからコンソールを開いてみましょう。小さなウィンドウでコンソールが開いたはずです。

ただし，この状態だと設定されたコンソールサイズは元のままでウィンドウだけ小さくなっています。DIRなどを取るといちいちスクロールバーを動かさねばなりません。そこでウィンドウサイズとコンソールサイズを一致させましょう。コンソールの表示アイコンから「行数を画面サイズにあわせる」を選択し（好みにより「桁数を画面サイズにあわせる」も），コンソールの環境メニューから「キー定義と環境を保存」を実行します。

これでコンソール.ENVが書き換わりました。シャーペンからちゃんと小さなコンソールが開くはずです。

あとはメニューメンテでメニュー2番のデスクトップアイコン用メニューを書き換えます。コンソールの項を選んで起動オプションに先ほどと同様の設定を加えるだけです。さらに必要があればシャーペン.ENV同様にエディタ.ENVも変更しておいてください。

先日本屋でX68000関係の本を見ていて疑問に思ったのですがX68000スプライトのBGパターンはBGの使用する枚数によって定義できるパターン数が変化するわけですが，ここに定義したパターンはスプライト用に表示せたりBG用に表示させたり，両方できるのでしょうか。　静岡県　西田　健二

まず確認しておきたいのは，BGで使用されるパターンと通常のスプライトとして使用できるパターンは（パターンサイズが違うことがありますが），基本的に同じものであるということです。

BGを使用するとスプライトの定義枚数が減るといっても，BGで消費される部分がBG用に使用するパターン定義に使われているわけではありません。BGでも普通のスプライトパターンを指定して端から置いていくだけです。

BGを使うと通常はスプライトのパターン定義エリアとして使用されているRAMが，BG用スプライトパターンの配置情報エリアとして使用されるようになります。

ですから，スプライト定義を256個分行っておいてBG表示モードにしてみると，わけのわからない配置でスプライトBGが設定されるはずですし（パターンがマップデータとして使われる），逆にBGをフルに使用しておいて，定義されていないはずのスプライト番号を指定すると，これまたわけのわからないパターンのスプライトが表示されるはずです（マップデータがパターンとして使われる）。

これでわかることは，スプライトパターン定義用エリアとBGエリアは基本的に等価なものであり（というか同じもの），ハードウェア側にしてみれば使用目的など関係ないということです。そもそも仕様としてこれだけフレキシブルな構成を持っている以上，わざわざ制限をつける回路にするほうが難しいといえるでしょう。

さて，通常のゲームは256×256ドットの画面モードが多用されるわけですが，この状態ではBG 1枚分の仮想画面は表示画面の4倍の大きさがあるわけです。つまり実際の画面の3/4が無駄になっています。縦スクロールゲームや横スクロールゲームにしても，通常は仮想画面の半分も使わないというのがほとんどでしょう。

そのような場合，BGの使われない部分に定義されたスプライトパターンはどのように扱われるのでしょうか？

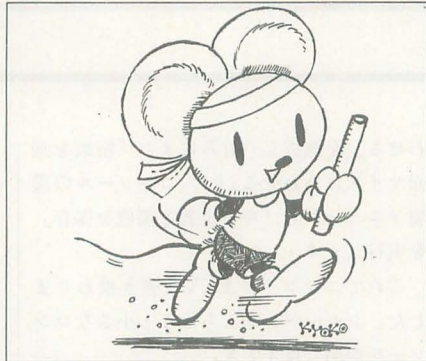
だいたい察しがつくと思いますが，見えていないBGはスプライトパターン用として使っても「誰も困らない」のです。

BGを使うとパターン定義数が激減してしまうので惜しいと思っていた人はうまくこの部分を活用してみてください。

問題となる点は，どの部分がBGに使われて，何番のパターンが定義できるのかということでしょう。これは使用状況によって変わってきますが，完全にスプライトパターンを初期化した状態から，使用する可能性のあるBGマップをすべてなにかで埋め尽くしてみればわかります。　（中野修一）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してください。

宛先：〒103　東京都中央区日本橋浜町3-42-3
ソフトバンク株式会社出版部
Oh!X編集部「Oh!X質問箱」係

FROM READERS TO THE EDITOR

虫の鳴き声が心地よく聞こえてくる季節です。皆さんはなにをして過ごしているでしょうか？　読書，音楽鑑賞，それともビデオ。でも，いちばん多いのはパソコンでプログラミングやゲームですかね。ただ，あまり夜更かしし しないように。

◆グラフィック，動画がよく特集として組まれ，進歩の度合も刻一刻という感じがし，興味あるところなんですが，なにせ“絵”の分野はメモリがたくさん必要なので，周辺機器の充実から入っていかなくてはなりません。本体だけで楽しめることっていまではなくなってしまったのでしょうか。まだまだ現役のX1をときどき触って，「たったこれだけのメモリでも結構おもしろいのに」と思ってしまいます。

三木　陽出(34)北海道

◆レンズフレアですが，あのようなフレアを光学ではゴーストと呼びます。また，レンズのデータがわかっていれば，ゴーストの出現する位置と大きさは求めることができます（レイトレーシングしなくても)。さらに通常，レンズには色がついてますから，反射の仕方によってカラフルなゴーストがでます（コーティングしているから)。「6角形のレンズフレア」といっているのは，丹さんが思われているとおり絞りの形です。つまり6角形ではなく6枚羽根の形です。

宮崎　圭介(25)徳島県

◆8月号の内容はちとわかりにくく感じられた。しかし，読み返してみればなんとかなりそうだったので安心した。　橋本　真樹(20)新潟県

難しいと感じる方も多くいらっしゃいましたが，自分でいろいろと試していくなかから，得るものがあるはずです。がんばってみてください。

◆アクセラレータの記事が復活しましたね。ところで完成の暁には基板を配布とありましたが，私としては完全キットもしくは，MPU,PLD,コネクタなど，地方では入手不可能な部品とセットにして配布していただきたいものです。工作記事を毎号楽しみにしていますが，地方では部品入手が極めて難しいのです。

鉄村　光太郎(27)北海道

本当にお待たせしました。完成にはもう少しかかりますし，実際に配布するとなると，いろいろな問題が出てきますので……。

◆「(で）のショートプロぱーてぃ」の「漢字」ものは，少し前のNHK「はるかなる漢字の旅」第1回に出てきた，漢字のみのアニメーションを思い起こさせてくれたので，2度笑いました。

加藤　洋介(27)埼玉県

ぱっと見てそれがなにかを認識させる漢字文化ってやはりすごいですね。

◆ただいま8月号を読みながら書いています。お，17ページの「HELL HOUND」はすごい。少し嬉しくなってきました。あ，22ページに「餓狼伝説SPECIAL」が……フフフフフ。あ，23ページ下に「VIEW POINT」が……ホホホホホ。あ，新作情報に「プリンセスメーカー」が……ヘヘヘヘヘ。あれ，その上に，まさか，ス，ス，スーパースト……ウヒャウヒャウヒャ（気が変になりそうです)。ああ，二浪してなければ……(現実に戻される)。　尾形　敦(19)東京都

9月号のゲームレビュー1本から今月号は一気に4本。この調子でガンガン新作が出てほしいものですね。

◆X68000を初めて手にした日を夢見ました。あのときめきを思い出して……そして目覚めたとき，私は自分が泣いていることに気づきました。あのときのことを思い出しながらX68000を分解し，内部をキレイに掃除して，これからもどうか一緒にがんばっておくれと呼びかけます。あのころのときめき，忘れたくはないですね。

和田　哲也(24)東京都

いつまでもそんな心を忘れないでいたいものですね。

◆いつも思っていた。「この内容で600円は安い，安すぎる……」。そして，9月号から680円になる。「まだ安いなあ。もっと高くしてもいいよん」と私は思う。　大久保　明弘(22)岩手県

ご理解をいただき大変ありがとうございます。これからもご期待に応えるためにも精進させていただきます。

◆毎月18日を心待ちにしていますが，本屋に行くとカラフルなほかのパソコン誌の色調に埋もれる感じで，目立たず「発売日なのにナイ」と店の人に尋ねるところでした。いいデザインなのにねぇ。　久保　英宣(23)三重県

今度からは目立つところに置き直しておいてくださいね。

◆私の知っている人は，「486-DX2 66MHzがオソイ！」と売りとばし，「486-DX4 100MHz」の機械を買いました。私はX68030の25MHzのスピードが速すぎて，ついていけません。文化の違いなのでしょうか。でも，33MHzのX68030がちょっぴり気になっています。大取　伸一(25)岡山県

マシンそのもののスピードよりも結局は動かすソフトが問題になるわけですから……。でもその友人もリッチですね。

◆「040turbo」買いまして，温度計も買いました。57℃，まだまだ大丈夫。

皆川　明久(23)奈良県

でも57℃か，触るのはちょっと……。

◆表紙の左下のソフトバンクのマークが成年コミックという文字に見えてしまう……。こ，これは，もしかしてヤバイのでは？

露崎　秀明(22)千葉県

いつもそんなものばかり読んでいるんですか？

◆その1　中古特価1,000円の横で同じソフトを新品特価500円で売る店（オイオイ)。
その2　ドラキュラFCディスク版を買ったら中身がディープダンジョンだった（友人の話ですけど)。その3　ついに登場！　中古ソフトを

◀岡村　直也　兵庫県
今月は7・8話と2枚の掲載です。その理由は編集部で続きが早く読みたいから。岡村さん，次を楽しみにしてますよ。

◀岡村　直也　兵庫県
そんなにぎっしりといったいなんのソフトが詰まっているんでしょう。でも，ここで注目はやはり恵子の腕の「X」の刺青ですね。

▲板橋　芳則　福島県
タイトルが「透明銃を持つ女」ということで、誰かに追われているのでしょうか。この女性、服は着ていないみたいです……。でも、どうやら

▲今井　健一　奈良県
てっきり、最初は「サムライスピリッツ」の移植希望かと思ったら、「ヴァンパイア」だったんですね。なんか歯茎から血が出そう……。

定価の２倍で売る店！　久米　豊信(26)大阪府

8月号に書いたときは冗談のつもりだったのですが，本当にタイトルと違う中身が入ったソフトを売っているなんて……。

◆生活費を5,000円に切り詰めているいま，Oh!Xの出費は痛すぎる。　小林　満幸(19)宮城県

これって食費とかも含まれているのでしょうか。それに，まさか１カ月分？

◆パソコン通信を始めた。料金が怖くて有料NETにはアクセスしてないが，草の根NETでも結構楽しい。なぜか絶対数の少ないはずのX68000ユーザーがPC-9801ユーザーを圧倒しているので，こんなところでもX68000ユーザーの積極性を感じました。もちろん私も積極的に書き込んでいますよ。　千葉　浩貴(21)宮城県

草の根NETと油断していると，知らぬ間に電話代が……。

◆８月号40ページの岡本訓さん，「アリストテレス」ではありませんが，シャープ最初のポケットコンピュータPC-1211は「ピタゴラス」という名前でした。Oh!Xの読者は何人ぐらい覚えているのだろうか？　鈴木　健児(23)愛知県

ということは，時代でいえば「ピタゴラス」から「ニュートン」へ一気に飛んで「ガリレオ」で少し戻るといったところですね。そうすると，「ガリレオ」の性能は……？

◆ううむ，Z-MUSICのインストール方法がよくわかりません。ハードディスクを手に入れてからはHuman68kの使い方など勉強したのですが，Z-MUSICのマニュアルは専門用語（パスを通す，環境変数など）が多すぎて……。初心者に詳しく教えてくれると助かります。SX-WINDOWを使い始めてから，130Mバイトのハードディスクの残り容量が少なくなり，そろそろもっと大きなハードディスクがほしくなってきました。安くならないかなー。　西山　浩(19)東京都

初心者の方にはなかなか難しいことも多いと思いますが，上に挙げてある２つの言葉も，Human68kのユーザーズマニュアルには解説してあるのでがんばってみてください。また，一般的なことについてはMS-DOSの解説書などでも参考になるので目をとおしてみてはどうでしょうか。

◆夏はやはりバイクで風を切って走るのが気持ちがいい。８月19日は「バイクの日」だそうですね。ライダーの皆さん，愛車でちょっとしたツーリングでもいかがでしょうか。

敏森　健裕(21)兵庫県

「バイクの日」とは知りませんでした。その日の私は友人と俳句の会を開いていました（大ウソ）。

◆８月号の「STUDIO X」に載った小笠原の友人Mです。ちゃんとOh!Xは毎月買っております。ご安心ください。しかし，私のX68000はゲームマシンと化し，いまは，仕事での使用マシンはPC-9801＋WINDOWSアプリに浮気してしまいました。彼にイヤミをいわれそうだ。

山沖　正直(30)神奈川県

しっかりとOh!Xに目をとおされているようで，とりあえず安心しました。休みの日にはゲーム以外のことにも使ってあげてくださいね。

◆ウチのX68000はバリバリと仕事をしています。ウチは自営業ですが，もはやX68000なしでは，従業員の給料も払えないほど。ウチの店の重役です！　X68000よ，よく遊び，よく働け！

市川　博基(18)愛知県

よく働いたX68000には，毎月の給料の代わりに，装備の充実が待っているんでしょうね。もし，充実されないとそのうちストを起こすかもしれませんよ。

◆私は悩んでいる。SCSIボードを買おうか，CDプレーヤーを買おうか……。どちらがよいでしょうか？　SCSIボードは去年の10月から，CDプレーヤーは去年の８月から夢を見ています（もう買っているかも……）。

西浦　宏和(20)愛知県

かれこれ１年間も夢を見ているわけですが，その夢の中ではCDプレーヤーやSCSIボードが追いかけてきたり……。

◆この年になって，おたふく風邪をひいてしまいました。しかも暑い盛りに。楽しみはOh!XとX68000だけです。そういえば，自動車のナンバーで42-19はないですね。ほかにも，もっとあったと思いましたけど。ちなみにひらがなでは，お，し，へ，というのが使われてません。

佐藤　泰満(35)宮城県

◆ガソリンスタンドで働いているのですが，常連客のなかには，ナンバーが42-19（死に行く）と49-89（四苦八苦）という車があったりします（実話）。42-19のナンバーを初めて見たときには「このナンバーって本当にあるんだ」と感動してしまいました。　境　秀行(23)福岡県

さて，どちらが本当なのでしょうか。あと，そういえば製造番号にも注意が必要かもしれません。友人が42XXという番号で事故を起こしたことが……。

◆珍しいことに金と時間に少し余裕ができたと思ったら，部屋の余裕が完全になくなってしまいました。やっぱり世の中はよくできてるとつくづく感じてしまいました。

藤原　彰人(24)岡山県

金と時間に余裕がないのに部屋にも余裕がない私の立場は……。

◆ご存じだと思いますが，高松の水不足はすごいことになってます。夏の一大イベント「高松まつり」が中止になりましたが，実に20年ぶりのことだそうです。いつまで続くんでしょうか。ちなみに職場のクーラーは水冷式ということで止められています。　早野　哲也(24)香川県

皆さん汗だくになって，お仕事をされていたのでしょうか。どうもご苦労さまです。

◆彼女ができた。とりあえず３人目。しかし，いまの子とつき合い始めてから，２人の子に告白された。おかげで１カ月ぐらい悩みまくって，まともに寝ていない……。これっていいことなのでしょうか……？　江城　憲之(18)大阪府

若いうちはたくさん悩んでください。あと，外を歩くときは背後にくれぐれも気をつけて……。

◆食中毒にかかりました。３日間，下痢と高熱に襲われ寝込む始末。原因は，賞味期限の切れた牛乳。冷蔵庫から出しっぱなしにしておいたのがいけないそうで。皆さんもお体を大切に。「教訓，怪しい食品は容赦なく捨てよう」。

金子　直史(18)新潟県

冷蔵庫に入っていた賞味期限の切れた牛乳をとりあえず捨てました。食中毒を未然に防ぐことができたかもしれません。でも，賞味期限内に食べてしまうのがいちばん。

◆このまえ，あまりプログラミングをしない友人と関数の返り値がどうのこうのという話をしていたら，「返り値と聞くと返り血を思い出して変になるからやめてくれ」といわれた。変なのはそっちやー！　浪越　孝宏(21)兵庫県

その友人には触れてもらいたくない過去があるのかも……。

◆このあいだ，友人（女の子）に「アンパンマンに似てる」といわれた。これって喜んでいいのかなぁ？　小山　優一(20)東京都

正義の味方に似ているのですから，思いっきり喜びましょう。贅沢をいってはいけません。それに小さい子供にはモテモテかも。

◆会社や寮で愛想のない機器ばかり相手にしているせいか，“ふわふわ・もこもこ”な小動物を

相手に遊びたいという衝動にかられることがある。ああ，動物が飼えればなあ。
渡辺　久孝(27)大阪府

“もこもこ”な小動物は想像がつくのですが，“ふわふわ”っていったい……。

◆須藤さん，“炎のコマ”なら知ってます。それに，「Mr.Do！」は温泉の旅館でハマッた覚えのある懐かしいソフトなので，絶対に買うつもりです。でも，あのころは小学生だったな（しみじみ）。
渡辺　英樹(19)大阪府

そういえば，深夜の某番組で「Mr.Do！ vs UNICORNS」を探していました。そのときはアーケード版の基板でしたが，マイコンソフトのパッケージもチラっと登場していました。

◆大学生になっていちばんの難関がテストであることがわかりました。特に理系の人はとてつもなく長いテストで，生きてる気がしないほどに弱っていきます。これなら，浪人のときのほうが楽だったような……。
青木　恭一郎(20)東京都

テストはこれから何度もあるし，さらに先には卒論もしくは卒研という大きな壁が待っていますよ。

◆この夏，暑くてふんばりがきかない。年をくったようだ……。
坊農　誠(23)福井県

お互いさまです。

◆どうも。8月号USER'S WORKSコーナー掲載の「HELL HOUND」の作者です。このたび，某地方局にアナウンサーとして就職することが決まりました。某地方の皆さん，これからはたぶん全国ただひとりのX68000ユーザーのアナウンサーとしてX68000の普及に努めますので，どうぞよろしくお願いします。
小松　泰郎(22)東京都

普及活動のレポートも楽しみにしています（冗談）。

◆5インチディスクを増設して「今後はメディアコンバートが楽になりましたよ」と同室の女性に教えたら，「うれしー，大きいのを3.5インチに入れたいと思ってたのよー」といわれた。で，持ってこられたのが8インチ……。でかすぎるよー。
松尾　和浩(33)新潟県

いまどき8インチを使えるマシンっていったい？

◆試験が終わったというのに，実験のために朝から夕方まで暗室にこもりっきり（泣）。空気もこもるし，暑いし，眠くなるし（密閉するから酸素が……），目も悪くなりそう。助けてー，ポパイ（錯乱）。ちなみに心理学（閃光）の実験です。
西尾　昌人(20)愛知県

ということは1日中暗闇のなかで生活しているわけですね。そのうち日の当たるところでは生きていけないかもしれませんね。

◆ようやく内定がもらえた。これからは卒業することに専念できる。パソコンもいままで以上にいじれる。人生最良の日だった。涙が止まらない。
松本　高佳(20)大阪府

無事に卒業できますように。

◆パソコンを使いたいので，ブレーカーが落っこちないように，クーラーを消している。あっつい……。たぶん編集室って涼しいんだろうなー。
渡辺　久理子(22)富山県

寒い……。

◆やっとTeXの本が出ましたね。私には，TeXを素直に「てくす」と発音して友人に「おめー，そりゃ“てふ”っていうんだ」と，思いっきり馬鹿にされたという悲しい思い出があります。
一ノ瀬　宣彦(23)東京都

ほかにも「てっく」といったりもするので，友人を「ふふん」と鼻で笑ってあげましょう。

◆ツインタワーの間のホコリをとった。数年間のホコリがとれて，とてもきれいになりました。
関　喜視(21)神奈川県

本体カバーを開けるともっとすごいホコリが待っていることでしょう。でも，カバーをとると正式な保証がたぶん受けられなくなるので……。

◆ペットの鳥が俺を見ている……。
友菊　学(17)千葉県

食べられないように気をつけてください。

◆奨学金がもらえる身分になった。衝動買いしそうで怖い……。
太田　崇貴(23)岡山県

なーに，少しくらいの借金，どうにでもなるものです（本当か？）。

◆2つも初体験をしてしまいました。ひとつ目は，ハンダごてなるものを使ったこと。思っていたより，難しいですね。ハンダが隣とくっついちゃったり，ビニール線をむいて芯を出すのに四苦八苦しました。でも，まがりなりにもできあがったときはとっても嬉しかったです。もうひとつは，ゲームセンターに連れて行ってもらったことです。8月号で須藤さんが書いていましたけど，私が小中学生のころは“ゲーセン”は不良の行くところだったので……。うっ，年齢を感じるなぁ……。でも，巷で評判の運転するアレや，格闘するアレはしませんでした。なにしろ反射神経をどっかに置き忘れてしまった人間なので……。
浅井　美雪(26)鹿児島県

反射神経はきっと母の体内に忘れてきて，弟か妹がその反射神経を受け取っているんじゃありませんか。

◆部活で大島に旅行に行きました。友達が「FF VI」を1日（徹夜）でクリアするというので，生ける攻略本として付き合いました。当然眠くなってしまい，最後のほうは2人の会話が成り立っていませんでした。疲れた。
小野　考平(17)東京都

部活の旅行でスーファミ？　しかも徹夜。終わったときは2人とも生ける屍と化していたようですね。それにしても，いったいなんの部活？　怪しい。

◆バイト先で毎日ショーを行うのですが，ある日，CCガールズが来ました。楽屋入りするところを裏口（調理場）から1mくらいの距離で見ました。思わず「うわっ，動いてる！」といってしまいました。
中島　貴史(18)滋賀県

うーん，その気持ちよくわかります。

◆ノーベル賞級の発明だと思ったモノ，それは「形状記憶シャツ」。私のような独身者にとって，まさに欠かすことのできないアイテムといえよう。
鹿又　健(25)東京都

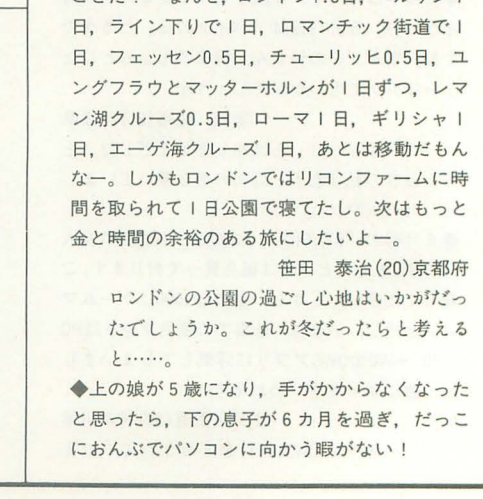
次は形状記憶スーツが登場して，丸洗いOKになり（なんか学生服みたい），クリーニング屋さんのピンチかも……。

◆ない金を作って行ってきた，ヨーロッパ縦断2週間の旅（ユーレイルパス使用）。もう，とっても忙しい忙しい。まさに駆け抜けたとはこのことだ！　なんせ，ロンドン1.5日，ベルリン1日，ライン下りで1日，ロマンチック街道で1日，フェッセン0.5日，チューリッヒ0.5日，ユングフラウとマッターホルンが1日ずつ，レマン湖クルーズ0.5日，ローマ1日，ギリシャ1日，エーゲ海クルーズ1日，あとは移動だもんなー。しかもロンドンではリコンファームに時間を取られて1日公園で寝てたし。次はもっと金と時間の余裕のある旅にしたいよー。
笹田　泰治(20)京都府

ロンドンの公園の過ごし心地はいかがだったでしょうか。これが冬だったらと考えると……。

◆上の娘が5歳になり，手がかからなくなったと思ったら，下の息子が6カ月を過ぎ，だっこにおんぶでパソコンに向かう暇がない！

◀近藤　隆生　埼玉県
この少女はイルカに乗っているのでしょうか。そういえば，どこかでシャチの背中に乗れるという企画があったような気が……。

◀岩瀬　貴代美　福岡県
この女性が左手に持っている封筒は，やはり好きな人へのラブレター？　ところで，弟さんとの野球観戦には時間に遅れずに行けましたか？

石川　栄一(35)新潟県

だっこはともかくおんぶをするのなら，ねんねこでしょってパソコンに向かえばOKです。これで英才教育もばっちりですね。

◆「ワニワニパニック」は僕のいちばん得意なゲームです。すでに1万円以上貢いだので，ワニのアルゴリズムを見切り，常に95点を超えることができるようになりました。調子が悪くても93点くらい，3回に1回はノーミスで99匹叩き切れたりします。ふっふっふ。でも，「悪魔城ドラキュラ」のドラちゃんが倒せなかったりします。ちくしょー，変身するなー！

嶋村　謙(21)大阪府

ドラちゃんを倒したあとは，カニに挑戦ですか？

◆先日，サイフを落としたことに気づきあせっていたら，電話がかかってきた。

相手：「サイフを拾ったんですけど」

自分：「どうもありがとうございます！」

相手：「お金はないけど，免許，カード類はあるみたいです」

自分：「いやー，助かりました」

相手：「落としたところを見て，すぐに追いかけたんですけど間に合わなくて……」

（以下省略）ガチャッ。

ん？　なんかナットクいかないぞ？

今田　智宣(20)兵庫県

落としたところを見られた人は，実はお金を抜き取った人だったのかも……。

◆女性の就職難についての発言に関して暴言をはきたいと思います。マスコミなどに出てくる女性たちは，女性を差別しているといいますが，私の知人（男性）は就職に失敗してしまいました。彼は自分の力が足りなかったのだといっています。女性たちのなかには，自分自身の能力のなさに気づかずに，女性だから失敗しているのだと思っている人がいらっしゃるのではないでしょうか。　清野　一男(23)山形県

このところの就職戦線は本当に大変ですね。ある会社では，「暑いですね。上着を脱いでください」といっておいて「いま脱いだ人は帰ってけっこうです」とぬかしたそうな。こんなのでは選ぶ側にも問題が……。まだ決まっていない人はがんばってくださいね。

◆「マッドストーカーX68」も「あすか120%」もCONFIGで「MISEROYO」と入力，ゲーム中，ファンクションキーや，ROLLUP，ROLLDOWNでいろいろできるようです。

山口　貴史(19)埼玉県

皆さんも試してみてくださいね。

◆8月号の「microOdyssey」を読んだあと，いろいろ考えましたが，やはり自分の体は自分で守る以外に方法はないのかもしれません。社会でどんな政策がとられたとしても，結局それを実行する1人ひとりの意志がちゃんとしていないとまるで意味がありませんよね。私もいまの身分からして規則正しい生活を送ることは不可能でないはずなのですが，つい先日も学会原稿の締め切りのせいで睡眠時間を削ってしまいました。反省。　宮本　明人(24)北海道

もちろん自分の体を自分で守れれば問題はありません。ただ，社会的にそれが認められているかいないかで1人ひとりが必要とする意志の強さは大きく変わってくると思いますが……。

◀青木　一師　奈良県
付録ディスクにぴったりのもみじのバックをありがとうございます。個人的には鹿が出てくると嬉しいのですが，女の子が可愛いからいいか。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目（売る・買う・氏名・年齢・連絡方法……）を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★パソコンクラブ「OREGA」は平成6年8月1日をもちまして休会いたしました。それに伴いまして，会報「俺が悩まないかんのか!?」も休刊いたしました。平成2年10月より皆様のおかげでいままでやってくることができました。ありがとうございました。さて，会員の方に連絡ですが，vol.13が届いていない方は11月末日までに連絡をください。それ以降はサポートできません。本当にお世話になりました。またどこかでお会いしましょう。〒910　福井県福井市文京4-9-5　メゾン山本201　新海敏之方「OREGA」サポート係

売ります

★システムサコムのSCSIハードディスク「Mocking Bird」(白)520Mバイトとサンケンの無停電電源装置「MPS-500」をセットで59,000円（送料込み）で売ります。どちらも新品同様で箱，説明書，付属品すべてあります。連絡は往復ハガキでお願いします。〒408-03　山梨県北巨摩郡武川村上三吹345　興石　学(24)

★X68000XVI用2Mバイト増設RAMボード「CZ-6BE2A」を25,000円（送料込み）で売ります。箱，説明書，付属品すべてあります。連絡は往復ハガキでお願いします。〒799-26　愛媛県松山市福角町625-8　加藤　和人(18)

★X68000XVI用2Mバイト増設RAMボード「CZ-6BE2A」を22,000円（送料込み）で売ります。箱，説明書，付属品すべてあります。連絡は往復ハガキでお願いします。〒660　兵庫県尼崎市東本町3-84-2　堀江　敬三(24)

★X68000CompactXVI用2Mバイト増設RAMボード「CZ-6BE2D」を20,000円で売ります。連絡は官製ハガキでお願いします。〒061-13　北海道恵庭市恵み野西1-6-7　鈴木　貴智(18)

★東京システムリサーチの「Xsimm10」と増設メモリ4Mバイト×2枚（70ns）を42,000円（送料込み）で売ります。箱，説明書つきで新品同様です。連絡は往復ハガキでお願いします。〒146　東京都大田区池上1-23-10　金井　利成(21)

★YHPのプリンタ「DeskJet 505J」とカラーキットのセットを20,000円以上で売ります。箱，説明書，付属品すべてあり，新品に近い状態です。連絡は往復ハガキでお願いします。〒344　埼玉県春日部市緑町2-4-10　森山　裕史(20)

買います

★SCSIボード「CZ-6BS1」または「SX-68SC」を送料込みで8,000円で買います。箱，付属品が揃っているものを探しています。連絡は往復ハガキでお願いします。〒719-01　岡山県浅口郡金光町大谷886-1　加藤　雅浩(25)

バックナンバー

★Oh!Xの1990年1～8月号をセットで10,000円（送料別）で買います。切り抜きは不可です。連絡は往復ハガキでお願いします。〒339　埼玉県岩槻市本町4-7-12　浜島　順一(20)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は8月号の内容に関するレポートです。

●8月号の特集「Graphic Movement」は，X68000のグラフィック能力を最大限に活用するための工夫が特集されていると考えていました。が，実際に特集を読むと内容的に少し横道に外れたように思います。「レンズフレアのシミュレーション」はグラフィック能力を生かした記事だったように思いますが，そのほかの特集記事はどちらかというとグラフィック機能（とメモリ）をぜいたくに使っているだけ，というような感覚を受けました。
壁谷　喜嗣(35)　X68000 EXPERT，PC-9821As，9801NS/E　愛知県

●SX-WINDOWにEX-WINDOWと，X68000のグラフィック環境もユーザーによる土台が完成しつつあるように思います。Z'sSTAFFやMATIERなどの優れたグラフィックツールで絵を描き，EX-WINDOWのフィルタで加工。そして，SX-WINDOWで表示。各種フォーマットの利用ができ，それを加工するノウハウも溜まってきたのはいいのですが，X68000のハードウェア自体に不満があるのも事実です。ウィンドウ環境を生かすためのメモリやCPUパワーの不足。グラフィック能力も8年前はぶっ飛んでいたとはいえ，いまではそれほど……という感じがします。
渡辺　祐介(19)　X68000　富山県

●最高のグラフィックを創り出すのに必要なものは，絵を描く才能や努力，創意工夫でしょう。画像フィルタはアイデアしだいで面白いことができて便利だし，高級感のあるCGを描くためには必需品です。しかし，画像フィルタなどはやはり必要に応じて，自分の手で即席で作ってしまうのが一番だと思います（X-BASICなんかでね）。Oh!Xとしては，いろいろなアイデアを寄せ集めて，アイデアだけたくさん紹介するといいかもしれません。
鈴木　朝夫(20) X1turboZ，MZ-731，PC-9801RA51，PC-8801VA，PC-6001mkII，FM-77AV，MSXturboR，ZX-81　神奈川県

●8月号に紹介されていた「X-SIMM VI」は，高く評価できる製品でしょう。X68000 XVIシリーズのメモリは，以前から高いだの品不足だのでなにかと問題になってきましたが，この製品のおかげですべて解消されたといえるのではないでしょうか。
　あと，ここ1年くらい石上氏のものをはじめ，突然アクセラレータの話題が多くなってきましたね。とりあえず，「H.A.R.P.」「HARP-FX」「040turbo」がサードパーティから発売（予定のものもあるけど）されました。高速化といえばCPUのクロックアップが主流ですが，実際どちらのほうがメリットがあるのでしょうか。私自身，030マシンを触るまでは68000，10MHzが重いと感じたことは，あまりありませんでした（「沙○曼蛇」とかは例外）。X68000以外にも，実際に386マシンや486マシン，RISCマシンなどを触ってみましたが，使用目的やアーキテクチャの違いが大きすぎて比較のしようがありませんでした。しかし，X68000 XVI以降高速機が出てきて数年。現在ではSX-WINDOWをはじめ，10MHzには少々荷が重いものも出てきました。重く感じるマシンからの高速化というものは，体感的に凄いものがありますよね。そういう意味で，030高速化より68000，10MHzの高速化に興味があります。
奥田　直也(21)　X68030，X68000 SUPER，ACE-HD，MSX2/PC-E550　愛知県

●8月号の特集を読んで，SX-WINDOWのテキストパレット1～7がちゃんとカラーとして使えれば，グラフィック画面があまり頼りなくてもそれほど困らないな，と思っています。Oh!Xで勝手に色を割り当てて使っちゃう，というのはどうでしょう（とてもやってほしい）。あとグラフィック圧縮展開ツールを作るならついでにライブラリ化してくれないかな，と常日頃考えていたので，PICSLICEライブラリはとても嬉しいですね。なにせ，GRROOT.Xのような便利なツールを持っていなかったので，6万色背景にするため，GL3ファイルをDOS _READしていたのです。記事の中では丹氏の「イメージ端子から入力すればどんな画像でも30fps出せる」の一文がいいですね。この言葉どおりならイメージユニットを持っている人なら，SX-WINDOWで「壁紙テレビ」とかが可能なのかな？　イメージ端子って12Vの電源だというくらいにしか考えていなかったけれど，実は結構侮れないところがあったですねえ。
石田　伯仁(21)　X68030，PC-9801VMII，PC-8801mkIIMR　神奈川県

●24ビットフルカラーが当たり前となりつつある現在，16ビットカラーは中途半端なクオリティなのかもしれません。しかし，いくら発色数が増えてもしょせん扱うのは人間です。つまり，ありあまる環境も整備されてはじめてその真価を発揮するものでしょう。その点で，X68000での65536色はバランスの取れたレベルであって，まだまだ実用に堪えうるものだと思っています。で，特集記事の中では「壁紙動画」が目をひきました。ただ，実際に動いているのが見られないのが残念ですが。そして，やはりくるべきときがきたか，と感じさせる80円の値上げ。現在，Oh!Xの読者が値上げと休刊の選択を迫られたらどういう答えが出るだろうか。X68000ユーザーにとってはかけがえのない存在であるだけに，休刊は非常に悲しいので，がんばって続けてほしいものです。
小林　佳徳(21)　X68000 XVI　新潟県

ごめんなさいのコーナー

8月号　ローテク工作実験室「基板を加工する」
P.89　X68030の取り替える抵抗が，R24，R246，R236となっていますが，R245，R246，R236の間違いです。R24は底面基板にはありませんので注意してください。ご迷惑をおかけしたことをお詫びします。

9月号　ステージデータ自動解法プログラム
P.61　最小ステップが報告されたハガキの整理ミスにより，プログラムを使って求めた最小ステップ数の28，36ステージの数値に誤った情報が紛れ込んでしまいました。正しくは，以下のとおりです。

28　11～13
36　20～42

9月号　X68030 D'ash
P.89　記事中に表記されていたX68030 D'ashの定価が誤っていました。正しい定価は350,000円（税込）です。誤った情報を載せたためご迷惑をおかけした関係者各位にお詫びいたします。

バグに関するお問い合わせは
☎03(5642)8182(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。
　また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

使うも遊ぶも あなたしだい フリスビーでも可

▶やってきました付録ディスク「もみじ狩りPRO-68K」。小物ツールを……のもくろみとは裏腹に，なにやら大きなファイルがごろごろと収録されてしまいました。もちろん，ただ単にサイズが大きいだけではありません。内容の充実度も保証付き。最新のSX-BASICはグラフィックエンジンのサポートにより，ずいぶんと面白いことができるようになりました。カードゲームにフィールド型RPGもどきに3D迷路など，収録されたサンプルたちを見ていただければ一目瞭然です。ただし，動作するSX-WINDOWはver.3.0以降ですので，まだバージョンアップがすんでいない人は，さっさとユーザー登録ハガキをシャープに送りましょう。

お次は，SLASHを使うためには欠かせないモデラMOD.Xです。前回よりも機能，操作性ともに格段に進歩しています。今回はちゃんとオブジェクトファイルもついているので，がんがん形状データを作成して，シェーディングされる美しいポリゴンを思う存分堪能してみてください。

そして，X-BASICゲームプログラミングユーザーに朗報のスプライト操作拡張関数。その名のとおりゲームを作るために便利な外部関数たちが盛りだくさん。この外部関数を使って，編集部をうならせる素晴らしいゲームを作ってみませんか？　パワーあふれる投稿をお待ちしています。

そうそう，「こいのぼりPRO-68K」に収録された「PUSH BON!」のオリジナルデータも多数（210ステージ分！）収録されています。ぜひ，遊んでね。

このほかにも，いろんなものが収録されていますので，特別企画をよく読んで「もみじ狩りPRO-68K」をお楽しみください。

▶「X68000マシン語プログラミング」は，著者多忙につきお休みとなってしまいました。

▶なお，今月号の特別定価が900円なのに，先月号の予告の予価を見てびびってしまった読者の皆さん。ごめんなさいね。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたフロッピーディスクを添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3
ソフトバンク出版部
Oh!X「㋢㋮㊔」係

SHIFT・BREAK

▶付録ディスクに入っているもののなかに，私がふだんローテクで使っている回路ライブラリがあります。もちろん，Easydraw専用ですが。ただ，このデータはディスクのマスターアップ時に突如入れることになったので，英語のものはスペルがいい加減だし，サイズもものによりけり。そのへんはちょっとだけごめんね。（瀧）

▶美術館の受付，交通量調査，後楽園の豆汽車運転手，スイミングスクールのコーチ，トラック配送助手，薬物実験ボランティア，ソフトハウスの外注，お歳暮の梱包など。そして，現在はビデオ屋の兄ちゃんだったりする。こうしてみるとやってないようでけっこーバイトしてたなーと妙に感心してしまう今日この頃。（いつ漫画家になれるのか？　の哲）

▶もう世間は涼しくなっただろうか？　ボクはこの夏をクーラーなしで乗り切ることになりそうである。部屋にないからしょうがない。それでも一応眠れる。コツは寝る前のジョギング。汗をかいてからシャワーを浴び，さっぱりしたところで一気に寝るのだ。体力もついて一石二鳥だけど，この水不足だと電気と水のどちらをとるか悩まされるなあ。（E.K）

▶一念奮起して部屋を片づける。貧乏性であることは自覚しているが，いちばん困ったのが買物でもらった袋の処分。「CD屋の袋はCDを入れるのにちょうどいいなあ」などと呟きながら，あれもこれも取っておくことにしたら結局部屋はもとのまま。部屋を片づけるよりも貧乏性克服のほうが先決だな，うんうん。（八）

▶今回の付録ディスクのために書いたCソースは軽く8000行を超えている。よくこれだけ書いたものだと自分でも関心してしまう。で，ごめんなさい，やっぱり外部コマンドはバグがあります。xclickを使うときはマルチビューにしないでください。マスターアップ後にならないと重大なバグが発覚しないのはやはり世の常なんだろうか。（けんと）

▶何年も前からほしくてしかたがなかったブレードランナーのオリジナルサウンドトラックがようやく出ました（大きな声ではいえませんが私は海賊版まで買いました）。いい。やっぱりサントラは妙なアレンジをしないのが基本ですね。ところで先月のpTeXの縦書きの件は私の事実誤認でした。事情は来月報告します。ごめんなさい。（A.T.）

▶実をいうと，先日，生まれて初めて家庭用ゲーム機を買った。メガドラ２だ。ぷよぷよ目的だったのだが，ぷよぷよバンドルの本体と２つ目のパッドとソフト２本を買って２万円強。これは安い。パソコンソフトだとこうはいかないもんね。でも，家庭用ゲーム機の限界も感じる。楽しいけど蓄積感がないのだ。この点ではまだパソコンのほうが上。（K）

▶今年の夏はMTBにほとんど乗れなかった。こんなにいい天気が続いたのに……。おまけに仕事が忙しくてRIEXの大会にも出られなかったし。その代わりにというわけではないが，何年かぶりに海へ行った。内房は波が穏やかだった。でも，男ばかり４人っていうのも……。そういえば白の水着はやっぱり透けるのではないだろうか？　ラッキー。（高）

▶深夜のタクシーで。「いままでお仕事ですか？」「雑誌の編集なんです」「っていうと，どんな？」「パソコン雑誌ですけど」「……ああ，それじゃ，私らが読むようなエッチなやつとは違うんですね」。いきなりエッチなといわれてもねぇ……。　深夜の車という密室の中で知らないおじさん相手では返答に困ってしまったあたしって，未熟者？（ふ）

▶付録ディスクに収録されたSLASH用モデラ「MOD.X」。僕も愛用しています。このモデラを使い始めてからというもの，モデリングがずいぶんと楽になりました。未完成だけど，坪井氏が精力的に制作に取り組んでくれているので，皆さんもなにか要望があれば遠慮なくお寄せください。できれば使ってあげてね。（PS.結婚しました。のＪ）

▶SX-WINDOWを使ってなくて，カラープリンタを持ってなくて，3D関係やゲームに興味がなくて，X-BASICを使わない人には評判が悪いだろうなあ。最近の調べによると読者の平均メモリ容量は5.1Mバイトだし，５月号のアンケートではSX関係やゲーム関係のリクエストがかなり多かったのは確かなのだが……。（U）

▶テレ・パソ・コンポなんちゅう広告を見ると，どうも六本木パソコンなるものを思い出してしまうのですが……。最近はテレビを見られるというのが，入門者向けパソコンのはやりのようですけど，世界初のパソコンテレビXIが登場したのが1982年の秋。VisionやWoodyやFreshなんかはXIのひとまわり違いの弟ってとこですか。（T）

microOdyssey

7月某日。結婚式場へ行く。まず，披露宴の内容を決めるためどんなオプションサービスを行うかと聞かれるたびに，ひたすらいりません攻撃。なにが楽しくてゴンドラに乗ったり，ドライアイスをたかなきゃあかんのか，と思いつつ，彼女と一緒に首を横に振りまくり。需要があるから存在するのだろうが，どんな人たちがこういったサービスを利用するのか，ちょっとだけ興味があった。あと，最低限やらなければ披露宴の間がもたない，とされるキャンドルサービスもかったるいのでヤメ。都合よく会場が空いていたため，すんなり式の日程も決まり，申し込みにかかった時間は2時間ほど。あまりにも簡単に事が運ぶので，なんだか拍子抜け。

8月某日。結婚披露宴の司会を担当してくれる人と打ち合わせ。やはり，ないないづくしで時間がもたないと司会者の人にいわれる。結局司会者の提案により，お色直しのあとのキャンドルサービスの代わりにお菓子を配り歩くことと，ビンゴゲームで間をもたせることにする。あとは出席者の中から乾杯の音頭をとってもらう人と，ビンゴゲームの罰ゲームとしてスピーチ，当日用意してあるカラオケを歌ってもらえそうな人を決めて，打ち合わせはおしまい。

8月某日。あっというまに結婚式当日。彼女とその母親と一緒に式場へ向かう。彼女は着付けのため美容室へ。僕も0次郎と化している髪の毛をまとめてもらうために美容室へ……と思ったら男性はダメとのこと。げげっ最初聞いた話と違うじゃないかと思いつつ，しょうがないので近場にある美容室を聞いて，わざわざブローしてもらいに行く。

そのあとは，無事着付けも終わり，写真撮影が行われた。そこで，カメラマンの指示を聞くたび，自分が人体モデルの人形になった気分になる。思わず込み上げる笑いをこらえるのにひと苦労。で，写真撮影が終わったあとは控え室へ。この頃になると人々も集まり始め，すでに見せものパンダ状態。定番の挨拶大会と友人たちが撮りまくる写真にひきつった笑顔で対応しつつ，挙式を待つ。

挙式はというと……はっきりいうとよくわからなかった（ここで大失態を演じたがもう忘れた）。どんな意味があるのかわからないまま，いろんなことをやらされただけ。

面白かったのは挙式のあとに行われた親族を含めた写真撮影。彼女の兄夫婦に子供がいたのだが，まだ，小さな子供ということでじっとしていない。どうしようか，というみんなの不安をよそに，カメラマンは「大丈夫です。とりあえずそのままにしていてください」とのこと。そして，いざ写真を撮るときになって「は〜い，僕たちこっちだよ」のひと言に子供たちが反応した瞬間をパチリ。さすがプロ。妙なところで親族一同も感心していた。

最後はいわゆる披露宴。なんの問題もなく，僕自身もさしたる失敗をすることなく終わったのだが，披露宴の間じゅう笑いっぱなしだった。以前から結婚することは決まっていたが，式場が決定してからわずか1カ月で結婚式という現実がとてもおかしかったのだ。結局，現実味のないまま終わってしまった結婚式だが，念願の彼女のウェディング姿も見れたことだし，とりあえず幸せなんでよしとしよう。（J）

1994年11月号10月18日(火)発売

特集　冬期集中BASIC講座

・SX-BASIC用ゲーム開発キット/SX-BASICの活用
・XSPRITE.FNCを使ったゲーム作成方法

ショウレポート '94データショウ
新製品紹介　BJC-400J/X680x0 Develop&libc II
全機種共通システム
シューティングゲーム作成講座(4)

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートBI	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3209)0656
	渋谷	大盛堂書店	03(3463)0511
	池袋	旭屋書店池袋店	03(3986)0311
	八王子	くまざわ書店八王子本店	0426(25)1201
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	043(224)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

基本的に，定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は，Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

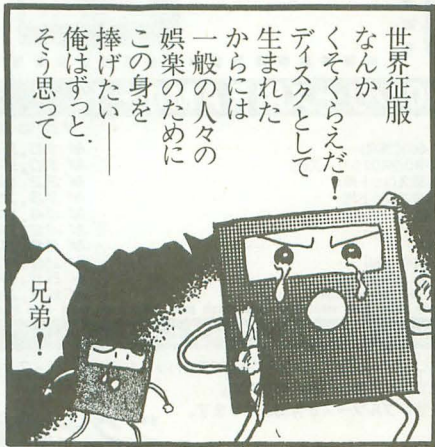
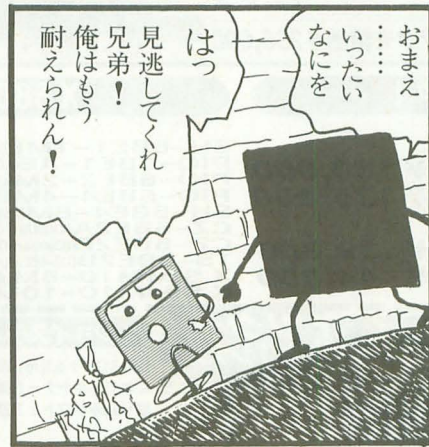
海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS（株）にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X　10月号
■1994年10月1日発行　特別定価900円（本体874円）
■発行人　橋本五郎
■編集人　稲葉俊夫
■発売元　ソフトバンク株式会社
■出版事業部　〒103 東京都中央区日本橋浜町3-42-3
Oh!X編集部 ☎03(5642)8122
販売局 ☎03(5642)8100 FAX 03(5641)3424
広告局 ☎03(5642)8111
■印　刷　凸版印刷株式会社

満開の電子(でんこ)ちゃん

作：いわい いっぺい
え：岡村祭

76号(8/18発売)は、レイトレ、アニメツールの大特集なのだ! 必見だよ。

講読方法：定期購読もしくはソフトベンダーTAKERUでお買い求めいただけます。
★定期購読の場合＝購読料第76号(94年9月号)より6ヶ月分9,000円(送料サービス、消費税込)を、現金書留または郵便振替で下記の宛先へお送り下さい。
　現金書留の場合：〒171 東京都豊島区長崎1-28-23 Muse西池袋2F ㈱満開製作所
　郵便振替の場合：東京　5－362847　㈱満開製作所
●ご注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
●3.5インチディスク版をご希望の方は、「3.5インチ版」とご指定下さい。
●新規購読の方は「新規」と明記して下さい。なお、特に購読開始号のご指定がない場合は既刊の最新号からお送りいたします。
●製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しします。
★TAKERUでお求めの場合＝1部につき1,600円（消費税込）です。
●定期購読版と内容が一部異なる場合があります。御了承下さい。
●お問い合わせ先　TEL(03)3554－9282（月～金　午前11時～午後6時）
（なお、定期購読版のバックナンバーについては定期購読の方のみご注文を承ります）

W・HOE
（富山県）

新たな時代に誘われて…。
みなさん、こんにちは。76号より2枚組になった電脳倶楽部には読者参加による読み物がたくさん掲載されるようになりました。教えたい人が教え、教わりたい人が教わるという、ひとつの理想的な形を作り出したと言えるでしょう。
さらに容量が増えて文章が図入りだったり、X68だけにこだわらない記事もあっていいですね。
また、自分のハガキやプログラムなどが掲載される恥ずかしさも、みなさんに味わってほしいです。
お、図書券届いたぞっ。

マイコン専門ショップ

P&A

SHARPエキスパートショップ

今が購入のチャンス!

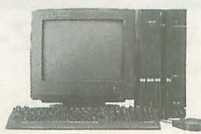
SHARP

9/17〜10/17

注目!!冬のボーナス一括払い手数料(金利)無料(平成6年10月末/11月末/12月末のいずれかをご指定下さい。)

X68030お買い得セット

(クレジット表：送料・消費税込み)

①X68030

- CZ-500C
- CZ-607D-TN(0.31mm、チューナー付)

定価合計¥497,800

P&A超特価¥299,000

12回	27,700	24回	14,400	36回	10,000	48回	7,800	60回	6,500

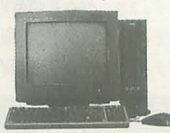
②X68030 HD

- CZ-510C
- CZ-607D-TN(0.31mm、チューナー付)

定価合計¥587,800

P&A超特価¥398,000

12回	36,300	24回	19,200	36回	13,300	48回	10,400	60回	8,700

③X68030 Compact

- CZ-300C
- CZ-607D-TN(0.31mm、チューナー付)

定価合計¥487,800

P&A超特価¥328,000

12回	30,000	24回	15,800	36回	11,000	48回	8,600	60回	7,200

④X68030 Compact HD

- CZ-310C
- CZ-607D-TN(0.31mm、チューナー付)

定価合計¥577,800

P&A超特価¥393,000

12回	35,900	24回	18,900	36回	13,100	48回	10,200	60回	8,600

■モニター変更の場合
- CZ-608D(B)…………に変更の場合¥ 3,000
- CZ-615D(チューナー付)に変更の場合¥56,000
- CZ-621D(B)…………に変更の場合¥64,000

加算して下さい。

旧シリーズ今が買いどき!!

送料¥2,000・消費税別

X68000 Compact XVI

(クレジット表：送料・消費税込み)

①

- CZ-674C-H
- CZ-607D(B)
- RGBケーブル

定価¥397,800

P&A超特価¥144,000

12回	13,200	24回	6,900	36回	4,800	48回	3,700	60回	3,100

②
- CZ-674C-H
- CZ-607D(B)
- RGBケーブル
- CZ-6FD5(FDD)

定価¥497,600

P&A超特価¥192,000

12回	17,500	24回	9,200	36回	6,400	48回	5,000	60回	4,200

X68000 SUPER-HD(HD80M内蔵)

③
- CZ-623C-TN
- CZ-607D(B)

定価¥597,800

P&A超特価¥144,000

④
- CZ-623C-TN
- CZ-607D(B)
- RGBケーブル
- CZ-6FD5(FDD)

定価¥697,600

P&A超特価¥192,000

■モニターの変更
- CZ-608D(B)…………に変更の場合¥ 3,000
- CZ-615D(チューナー付)に変更の場合¥56,000
- CZ-621D(B)…………に変更の場合¥64,000

加算して下さい。

(送料¥1,000・消費税別)

本体(単品)

⊙CZ-674C (Compact XVI) P&A超特価¥85,000

⊙CZ-623C (SUPER-HD) P&A超特価¥85,000

カラーイメージユニット

限定

- CZ-6VT1-BK

定価¥69,800

特価¥52,500

インテリジェントコントローラ

限定

- CZ-8NJ2

定価¥23,800

特価¥13,800

X68000/68030用 メモリボード

(送料¥700・消費税別)

■I/Oデータ
- SH-5BE4-8M(30用)…特価¥39,500
- SH-6BE1-1ME(600C用)…特価¥10,200
- PIO-6BE1-AE(ACE/PRO/PROII用)・特価¥10,200
- PIO-6BE2-2ME(拡張スロット用)・特価¥21,000
- PIO-6BE4-4ME(〃)・特価¥35,300

■シャープ
- CZ-5BE4(30用)………特価¥39,800
- CZ-5ME4(5BE4用増設)・特価¥36,500
- CZ-6BE2A(XVI用)……特価¥38,900
- CZ-6BE2B(XVI、674C増設)特価¥37,500
- CZ-6BE2D(674C用)…特価¥20,500

モデム&FAXモデム

(送料¥1,000)

〈インテグラン〉
- MP1414F(FAXモデム・ポケット型)………特価¥31,000

〈サン電子〉
- MS1414AVF(FAXモデム・ボックス型)……特価¥30,000

〈アイワ〉
- PV-PF144(FAXモデム・ポケット型)………特価¥29,000
- PV-AF144V5(FAXモデム・ボックス型)……特価¥29,000

〈オムロン〉
- MD-96XT10V(FAXモデム・ボックス型)…特価¥30,000
- MD-144XT10V(FAXモデム・ボックス型)…特価¥35,000
- ME1414B(FAXモデム・ボックス型)………特価¥22,000

〈マイクロコア〉
- MC14400FX(W)(FAXモデム・ボックス型)…特価¥23,000
- MC24FC5(W)(FAXモデム・ポケット型)…特価¥20,000

MO&CD-ROM

(送料¥1,000)

■CS-M120(コパル)
光磁気ディスク(X68000用)
●ケーブル、ターミネータ付
定価¥178,000
特価¥93,000

■LMO-FMX330TS
(ロジテック)●ケーブル付
定価¥168,000
特価¥97,000

MO
- UL-312E-S(緑電子)…特価¥ 62,000
- MO-120S(ICM)………特価¥ 88,000
- MO-230S(〃)………特価¥110,000
- LMO-340(ロジテック)…特価¥ 85,000
- LMO-400(〃)…特価¥110,000

(ケーブル別売)

CD-ROM
- CXA-301S(緑電子)…特価¥ 20,800
- CXA-450S(〃)…特価¥ 38,800
- CD-310S(ICM)………特価¥ 33,800
- CD-450N(〃)………特価¥ 39,800
- CD-600S(〃)………特価¥ 68,600
- LCD-550(ロジテック)…特価¥ 38,800

東京システムリサーチ製(X SIMM)

(送料¥700・消費税別)

(X SIMM VI)

⊙XVIシリーズ専用SIMM増設式メモリボード
- X SIMM VI(634C用)…定価¥16,500➡特価¥13,000
- X SIMM VIc(674C用)・定価¥16,500➡特価¥13,000

⊙増設SIMMメモリ(72PIN)
- 4MB(70ns)………………特価¥11,800
- 8MB(70ns)………………特価¥27,800
- 4MB(60ns、24MHz以上用)………特価¥16,500
- 8MB(60ns、24MHz以上用)………特価¥28,000

- 6MB(70ns、メーカー純正品)……特価¥31,000

(X SIMM 10)⊙SIMM増設式メモリボード
- X SIMM 10………定価¥18,000➡特価¥15,700

⊙増設SIMMメモリ
- 1MB×2……特価¥ 9,000
- 4MB×2……特価¥30,000

- 10MB例 X SIMM10+1MB×2+4MB×2‥¥54,700

X68000/68030専用ハードディスク

(送料¥1,000・消費税別)

外付

■ロジテック

⊙SHD-B240N-FMX(ケーブル付)(240MB、14ms、64K)
…………定価¥59,800▶特価¥45,000

⊙SHD-B340NU(ターミネータ、ケーブル付)
(340MB、12ms、96KB)…………特価¥49,800

■富士通

⊙HD-M260(モッキンバード)(260MB、14ms、256K)
…………特価¥39,800

⊙HD-K520(モッキンバード)(520MB、12ms、240K)
…………定価¥128,000▶特価¥69,800

■ジェフ

⊙GF-270(270MB、12ms、128K)
…………定価¥89,800▶特価¥59,000

⊙GF-540(540MB、12ms、128K)
…………定価¥128,000▶特価¥69,800

内蔵

■CZ-500C/300C専用

⊙CZ-5H08(80MB/23ms)
…………定価¥ 98,000▶特価¥71,800

⊙CZ-5H16(160MB/18ms)
…………定価¥135,000▶特価¥99,500

●価格は変動します。ご注文の際は必ずお電話で価格と在庫をご確認下さい。●本広告に掲載の商品には送料及び消費税は含まれておりません。

ズバリご奉仕

P&Aならではの 新品パソコン 5年保証

《業界No.1の"P&Aメンテナンスサポート"》
最高の保証システム

① 業界最長の新品パソコン5年保証
（※モニター・プリンター3年間保証!! ※一部商品は除きます。）
② 中古パソコンの1年間保証(※モニター・プリンター6ヶ月間保証!!)
③ 初期不良交換期間3ヶ月(※新品商品に限らせていただきます。)
④ 永久買取保証
⑤ 配達日の指定OK!!(土曜・日曜・祭日もOK!!)
⑥ 夜間配達もOK!!(※PM6:00～PM8:00の間 ※一部地域は除きます。)

便利でお得な支払いシステム

① 翌月一括払い手数料無料(ご利用下さい。)
② 業界No.1の低金利!!
③ 月々の支払いは¥1,000より
④ 9ケ月先からのスキップ払いOK!!
⑤ 84回までの分割、ボーナス併用OK!!
⑥ カレッジクレジット
⑦ ステップアップクレジット
⑧ ボーナスだけで10回払いOK!!
⑨ 現金一括支払いOK!!
⑩ 商品到着払いOK!!(代引き手数料が必要になります。10万円まで900円)
（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

●法人向け リースシステム
業務に最適なシステムを構築します。
損金処理が可能なリース契約をどうぞ。

周辺機器コーナー（送料¥1,000・消費税別）

カラーイメージスキャナ
■JX-325X 限定
定価¥190,000
特価**¥79,800**

カラーイメージジェット
■IO-735X-B
定価¥248,000
特価**¥128,000**

ビデオスキャナー
■CZ-6VS1
定価¥178,000
特価**¥135,000**

FDD(5インチ×2基)
■CZ-6FD5
定価¥99,800
P&A超特価
¥49,800

プリンター(ケーブル用紙付)
- MJ-500V2 (エプソン)…特価¥44,300
- MJ-1000V2 (〃)…特価¥64,300
- MJ-700V2C (〃)…特価¥78,300
- BJ-220JC (キヤノン)…特価¥58,000
- BJ-10V Lite (〃)…特価¥31,300
- BJ-15V PRO (〃)…特価¥39,700
- LBP-A404GII (〃)…特価¥99,500
- BJC-600J (〃)…特価¥78,300
- JET505J PLUS(YHP)…特価¥50,300

光磁気ディスク(X68000用)
■CS-M120(コパル)
●ケーブル、ターミネータ付
¥178,000
特価**¥93,000**

- CZ-6BV1………定価¥21,000▶特価¥15,900
- CZ-8NM3………定価¥ 9,800▶特価¥ 7,200
- SH-6BF1………定価¥49,800▶特価¥36,500
- CZ-6BP1………定価¥79,800▶特価¥57,000
- CZ-6BS1………定価¥29,800▶特価¥21,500
- CZ-8NJ2(限定)…定価¥23,800▶特価¥13,800
- CZ-6CS1(674C用)・定価¥12,000▶特価¥ 8,900
- CZ-6CR1(RGBケーブル)・定価¥ 4,500▶特価¥ 3,600
- CZ6CT1(テレビコントロール)・定価¥ 5,500▶特価¥ 4,400
- CZ-6BP2………定価¥45,800▶特価¥33,300
- CZ-5MP1(X68030用)・定価¥54,800▶特価¥42,000

送料¥700・消費税別

■システムサコムボード
- SX-68MII (MIDI) 定価¥19,800 特価¥13,500
- SX-68SC (SCSI) 定価¥26,800 特価¥17,500

X68000用ソフトコーナー（送料¥700・消費税別）

- Z's STAFF PRO68K Ver. 3.0(ツァイト) ……定価¥58,000▶特価¥37,500
- Z's TRIPHONYデジタルクラフト(ツァイト) ……定価¥39,800▶特価¥27,000
- マジックパレット(ミュージカルプラン) ……定価¥19,800▶特価¥14,200
- たーみのる2(SPS) ……定価¥17,800▶特価¥13,000
- サイクロンEXPRESS α68 ……定価¥98,000▶特価¥69,000
- Video PC for X680X0(マイクロウェアシステムズ) ……定価¥58,000▶特価¥46,400
- X WINDOWS V.11.5(マイクロウェアシステムズ) ……定価¥30,000▶特価¥25,500
- Double Book IN(計測技研) ……定価¥12,800▶特価¥ 9,600
- OS-9/X68030 V.2.4.5(マイクロウェアシステムズ) ……定価¥25,000▶特価¥19,900
- C&Professional Pack V.3.2(マイクロウェアジャパン) ……定価¥80,000▶特価¥57,800
- マチエール Ver.2.0 ……定価¥39,800▶特価¥28,800
- F-Calc for X68K ……定価¥14,800▶特価¥11,000
- CZ-214MSD SOUND PRO68K ……定価¥15,800▶特価¥11,300
- CZ-215MSD Sampling PRO68K ……定価¥17,800▶特価¥12,500
- CZ-225BSV Multiword Ver.2.0 ……定価¥32,000▶特価¥23,000
- CZ-227BS TOP財務会計PRO-68K ……定価¥200,000▶特価¥154,000
- CZ-243BSD CYBERNOTE PRO68K ……定価¥19,800▶特価¥15,000
- CZ-247MSD MUSIC PRO68K(MIDI) ……定価¥28,800▶特価¥20,500
- CZ-249GSD CANVAS PRO68K ……定価¥29,800▶特価¥22,000
- CZ-251BSD Hyperword ……定価¥39,800▶特価¥29,400
- CZ-253BSD CARD PRO68K Ver.2.0 ……定価¥29,800▶特価¥22,700
- CZ-257CSD Communication PRO68K Ver.2.0 ……定価¥19,800▶特価¥15,300
- CZ-261MSD MUSICstudio PRO68K Ver.2.0 ……定価¥28,800▶特価¥21,200
- CZ-263GWD Easypaint SX-68K ……定価¥12,800▶特価¥ 9,800
- CZ-264GWD Easydraw SX-68K ……定価¥19,800▶特価¥15,300
- CZ-265HSD NewPrint Shop Ver.2.0 ……定価¥20,000▶特価¥15,400
- CZ-266BSD PressConductor PRO68K ……定価¥28,800▶特価¥22,000
- CZ-267BSD CHART PRO68K ……定価¥38,000▶特価¥29,800
- CZ-271BWD EG-Word ……定価¥59,800▶特価¥44,900
- CZ-272CWD Communication SX68K ……定価¥19,800▶特価¥14,500
- CZ-274MWD MUSIC SX68 ……定価¥38,000▶特価¥29,300
- CZ-275MWD SOUND SX68K ……定価¥15,800▶特価¥11,500
- CZ-284SSD OS-9/X68000 Ver.2.4 ……定価¥35,800▶特価¥25,600
- CZ-286BSD BUSINESS PRO68K ……定価¥28,000▶特価¥20,500
- CZ-288LWD開発キット(workroom) ……定価¥39,800▶特価¥29,700
- CZ-289TWD 開発キット用ツール集 ……定価¥12,800▶特価¥ 9,600
- CZ-290TWD SX-WINDOW ディスクアクセサリー集 ……定価¥14,800▶特価¥11,500
- CZ-295LSD C-Compiler PRO68K Ver.2.1 NEW KIT ……定価¥44,800▶特価¥32,500
- CZ-296SS/SSC SX-WINDOWS Ver.3.1 ……定価¥22,800▶特価¥17,600

☆ゲームソフト25%OFF OK!!(一部ソフト除く)

全国通販 ★頭金なし! ★即日発送

- お近くの方はお立寄り下さい。専門係員が説明いたします。
- 本体単品で特価で受付します。詳しくは電話にてお問合せ下さい。
- ビジネスソフト定価の20%引きOK! TELください。

P&A特選 今月の中古特選品

- CZ-600C‥¥45,000
- CZ-601C‥¥45,000
- CZ-611C‥¥50,000
- CZ-652C‥¥55,000
- CZ-612C‥¥75,000
- CZ-603C‥¥65,000
- CZ-653C‥¥58,000
- CZ-612C‥¥70,000
- CZ-623C‥¥70,000
- CZ-674C‥¥70,000
- CZ-634C‥¥100,000
- CZ-644C‥¥145,000

※上記は単品価格、モニター別売。

新古品 限定
- CZ-674CH
- CZ-608DH
¥128,000

中古品
- CZ-674CH
- 68000専用モニター付
¥89,000

限定
- CZ-634CTN(チタン)(中古)
- CZ-613D(グレー)(新品)
¥180,000
(モニターをCZ-614TN(チタン)に変更の場合¥20,000加算)

中古品
- CZ-623C-TN
- 68000専用モニター
¥98,000

新古品 限定
- CZ-644CTN
- CZ-604DB
¥208,000

中古品
- CZ-644CTN
- 68000専用モニター付
¥178,000

高額買取り(新品もOK)格安販売

■まずはお電話下さい。
下取り専用 買取り電話▶ ☎03-3651-1884 FAX. 03-3651-0141
■下取り・買取りで、お急ぎの方は、直接当社に来店、または宅急便にてお送りください。

買取り価格…完動品・箱/マニュアル/付属品の価格です。中古販売…1年間保証付。

- 下取りの場合…価格は常に変動していますので査定額を電話で確認してください。（差額は、P&A超低金利クレジットをご利用ください。）
- 買取りの場合…現品が着き次第、3日以内に高価買取金額を連絡し、振込み、又は書留でお送り致します。
- 近郊の方はP&A本店に直接お持ちください。即金にて¥5,000,000までお支払い致します。

- 最新の在庫情報・価格はお電話にてお問い合せください。
- 買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せください。
- 価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認ください。
- 本商品の掲載の商品の価格については、消費税は、含まれておりません。
- 現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せください。

P&A特選 パソコンラック&OAチェアー(消費税込み)(送料無料、離島を除く)

※全機種→キャスター付 ※フレーム色:ホワイト
※上から2番目棚板移動可能(4段)
※3段の場合、上から2番目の棚板は付いておりません。

※上下2分割式/スライドマウステーブル、中棚板は2段階に可動します。
※フレーム色:グレー

①¥9,270
- 布張り ダークグレー
- ガスシリンダー

②¥11,330
- 布張り ダークグレー
- ガスシリンダー
- 肘付

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)
〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。●1回～84回払いまで出来ます。但し、1回のお支払い額は¥1,000円以上。
〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。(電信扱いでお振込み下さい。)

〔振込先〕さくら銀行 新小岩支店
当座預金 2408626 (株)ピー・アンド・エー

超低金利クレジット率

回数	3	6	10	12	15	24	36	48	60	72
手数料	2.6	3.5	4.4	4.9	7.8	10.4	14.4	18.9	24.4	31.8

株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目2番地20号
●営業時間:AM10:00～PM7:00 日・祭:AM10:00～PM6:00
☎03-3651-0148(代)
●定休日/毎週水曜日 FAX.03-3651-0141 MAC/DOS Vフロア ☎03-3655-4454

■姉妹店/ユーアイネット ギガ店 OPEN(千葉県浦安市入船1-4-1 ショッパーズ 1F ☎0473-81-2107)もよろしく!!

※お支払いは、便利な商品到着払い〈手数料(10万円まで900円)要〉をご利用下さい。

超級特価 X68030(CZ-500C)+040turboバンドルセット ¥328,000!

オリジナルアプリケーション開発速報

1994年10月1日発行（毎月1回1日発行）通巻13号 *FirstClassTechnology* 発行

SX-WINDOW用CD-ROM辞書検索ソフト

好評発売中

SX広辞苑《EPWING対応版》

標準価格 ソフト単体 ¥19,800

岩波書店「広辞苑第4版CD-ROM版」バンドルセット ¥43,800

あの、SX広辞苑がグレードアップして新登場！

SX広辞苑《EPWING対応版》は、この「広辞苑第4版CD-ROM版」を効率的に検索し、120％活用するためのソフトです。

●SX広辞苑《EPWING対応版》の特長

- 豊富でパワフルな検索方法により、必要な情報をすばやくピックアップ。
- 使う側に立って操作系をリニューアル。さらに簡単に、さらに鋭く作業を行なえます。
- 広辞苑の最新版である第4版をもとにしたCD-ROMを使用するので、よりコンテンポラリーなキーワードにアクセス可能です。
- SX-WINDOW上で動作するので記事の参照や引用がとても簡単。シャーペンやEGWordと組み合わせて活用できます。（ただし、広辞苑では大量の引用は禁止されています）
- シャーペンと融合して語句の検索を行なうシャーペン用外部コマンド"LightWing.X"を同梱。複雑な検索を行なう場合はSX広辞苑.Xを、普段よく使う単純な検索にはLightWing.Xを、という使い分けも可能です。
- 広辞苑第4版CD-ROM版と同様に、EPWING(V1)規約にもとづいたCD-ROMタイトルなら、ほとんどのCD-ROMの内容を検索できます。

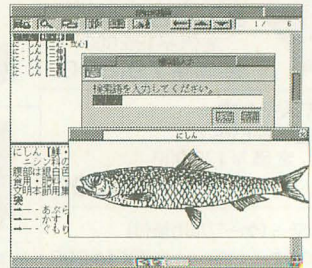
● 当社で動作を確認したEPWING(V1)タイトル

現在約20タイトル発売されているEPWING(V1)準拠のCD-ROMのうち、以下のタイトルについては当社で動作を確認しました。

なお、SX広辞苑《EPWING対応版》上での動作に関して、各タイトルの出版社に問い合わせることはご遠慮ください。

タイトル	出版社
・広辞苑第4版CD-ROM版	岩波書店
・岩波電子日本総合年表[EPWING版]	岩波書店
・CD-ROM最新医学大辞典[スタンダード版]	医歯薬出版
・漢和辞典漢字源[EPWING版]	学習研究社
・リーダース英和辞典	研究社
・三省堂ワードハンター　―マルチROM辞典―	三省堂

この他のタイトルについても動作確認作業を進めています。

●動作環境

- SX-WINDOW 3.0以上
- SX-WINDOW動作中の空きメモリとして1MB以上を推奨
- CD-ROMドライブ(CD-ROM Driver Ver2.0が付属するので、CD-ROM Driverを別途お買い上げいただく必要はありません。CD-ROM Driverのマニュアルや添付ソフト等は付属しません）

好評発売中

X680x0用Ether net接続パック

EthernetStarterPack/X680x0

発売記念特価 ¥78,000

ESP/Xは、Ether netアダプタ「Ether＋」と、TCP／IPドライバ、そして基本的なアプリケーションからなるパッケージです。

- Ether＋(米コンパチブルシステムズ社製)
  SCSIインターフェースを介してEther netとX680x0を接続するためのハードウェアです。
  ※ 10BASE-2対応モデル・10BASE-T対応モデルの2種類があります。
- TCP/IPドライバ
  X680x0でTCP/IPをサポートするドライバ。ソケットも利用可能です。
- 基本的なアプリケーション
  ftp、telnet(いずれもクライアント)等、基本的なアプリケーションを標準添付。ドライバを活用するためのライブラリも付属します。

●動作環境

- Human68k ver3.0以上
- メモリ常駐量500KB前後
- SCSIインターフェース内蔵機種以外はSCSIボードが必要

X68030用　68040搭載アクセラレータ

X68040turbo

標準価格 ¥98,000　ヒートシンク別売 ¥1,000

040turboは、68040を搭載したX68030(5インチタイプ)専用のアクセラレータです。040turboを装着することで得られるパフォーマンスは、従来の2～3倍！ 計算、特に浮動小数点演算中心のソフトならば、さらにそれ以上の高速化も望めます。

詳しくはソフトバンク刊「X68040turbo～A Story of Making "After X68030"～」(BEEPs著)をご覧ください。

040turboは当社のショップBASIC-HOUSEでの直販、および通販でのみお買い求めいただけます。ご注文いただいてからしばらくお待ちいただく場合もありますので、お早めにご注文ください。

バージョンアップのお知らせ!!

040Turboに一部不具合がありましたので、バージョンアップサービスをいたします。

症状：040モードでごくまれにバスエラーの処理でハングアップする場合がありました。

対象となる方：IC2に「V5」と印刷されている040Turboをお使いの方。

バージョンアップ方法：80円切手を貼り、住所氏名を記入した返信用封筒を同封の上、基板からIC2を抜き、封書でお送りください。IC2の不具合を修正後、お送りいたします。

所要期間は1週間程度とお考えください。

送付先：〒320 栃木県宇都宮市京町11-18 OYAMAビル2F First Class Technology 040turbo係

SX-WINDOW用スケジュール管理ソフト

DoubleBookin'

標準価格 ¥12,800

バージョンアップ!!

CD-ROM Driver Ver2.00

標準価格 ¥4,800

Ver.2.00となりSCSI-2専用となりました。

当社サポートネットTECOSYS-3でVer1.06からのバイナリ差分をダウンロードしていただけます。

SX-WINDOW 用Photo-CDビュアー

SX-PhotoGallery

標準価格 ¥15,800　通販特価 ¥15,000

X680x0用フリーソフトウェア集CD-ROM

FreeSoftwareSelection Vol.2

標準価格 ¥6,000

お求めはお近くのパソコンショップ、または弊社通販部（TEL：0286-22-9811）へお申し込みください。

通販ご希望の方は、ソフト代金＋送料¥1,000に消費税を加え、ご住所・お名前・電話番号・商品名を明記した紙を同封の上、現金封筒でお申し込みください。

低金利クレジット　通信販売送料 全国一律¥1,000 長期クレジット可能

※表示価格に消費税は含まれておりません

株式会社 計測技研　マイコンショップBASIC HOUSE 本社／ショールーム／通販部
〒321 栃木県宇都宮市竹林町503-1
TEL 0286-22-9811　FAX 0286-25-3970

TECOSYS-3は(0286)51-1430で24時間運転中! 当社製品ユーザーの方も、そうでないかたもアクセスしてね♪ CD-ROM Driverの差分もあるっす。(^_^)

クイーン・オブ・デュエリスト外伝α+

Ⓒ1994 アクミックス／蘭宮涼／あずまきよひこ／海野やよい／マイケル原腸／ここまひ／うたたねひろゆき／富士参號／龍炎狼牙

※このソフトは18歳未満の方はご購入できません。

¥9,800（税別）

企画・開発／アクミックス
X68000/X68030対応

QD vs 外伝ファン待望、夢の対決だ！

★マンガ家8人が登場キャラクターをそれぞれデザイン！
★総勢10人のキャラクターを自由に選択！
★QDキャラ（美由紀、エミリー、龍凰）がマイナーチェンジして外伝キャラと対決！
★QD＆外伝キャラクターの18禁スペシャルグラフィックで満足度◎！
★キャラクターパターンはさらに増え、よりリアルな動きを再現！
★背景もX68000用にすべて描きかえました！
★PCMもさらにパワーアップして、迫力のサウンドを保証！
★必殺技、連続技、キャンセル技等の要素も含み、戦闘バランスもパワーアップ！

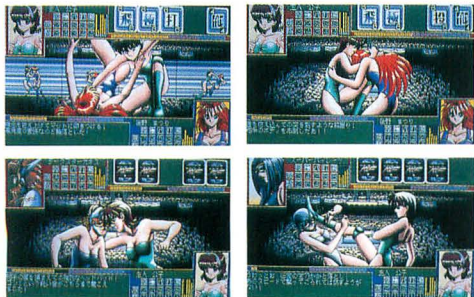
※画面写真はFM-TOWNS版です

目指すは最強団体！

X68000

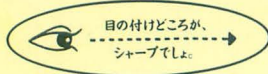
レッスルエンジェルス3

限定
パッケージ版
発売中

ゲーム業界初のプロレス団体経営シミュレーション！
新人、現役等約50名のレスラーが登場。
彼女の内、誰を引き込み、敵に回すかはプレイヤー次第。
最大3人までのマルチプレイが可能となり、そしてセーブデータ同士の団体対抗戦も加え、更に面白さが倍増した！
もちろん、ハイスピードなカードバトルも健在！
合体技等を加えた技はオールリニューアル。
これであなたもプロレス異次元ワールドに突入だ！

企画・開発／グレイト
X68000/X68030対応

¥7,800（税別）

株式会社エクシング TAKERU事務局
〒467 名古屋市瑞穂区苗代町2番1号
ブラザー技術開発センタービル2F
TEL(052)824-2493（受付時間：月〜金 13:00〜18:00）

営業所
東京営業所 (03) 5443-4967
大阪営業所 (06) 258-3024

感性を光らせる。

さまざまなフィールドで、研ぎ澄まされた感性に応える潜在能力の実証

X68の潜在能力は、まさに時代とともに証明されつつあります。
開発当初より、現在のマルチメディア環境を想定していた事実。
グラフィック能力はもちろん、ADPCM対応、オリジナルウィンドウシステム、
X68にとってこれらは、数年前のスペックなのです。
パソコンの存在そのものを革新した「創造性」、マインドを喚起する「こだわり」、
いま、先見のユーザーに支えられたX68は
そのコンセプトの開花を得て、多彩なフィールドへと飛翔します。

Workbench

WSとしての楽しみ

たとえば、リアルタイム・マルチタスク・
オペレーティング・システムOS/9。
X68030の能力を最大限に引き出す
UNIXライクな操作性と洗練された機能。
X-WINDOWや動画ツールのサポートで
さらに深い楽しみが…。

※OS/9はマイクロウェア・システムズ㈱の登録商標です。
※UNIXは、X/Openカンパニーリミテッドが独占的にライセンスする米国および他の国における登録商標です。

Create

創造するよろこび

SX-WINDOW開発支援ツールが
創造力を刺激する。
ソフト開発に必要なツールや
サンプルプログラムを多彩にバンドル、
ウィンドウ上で効率よく作業でき、
初めてプログラムに挑む人への
やさしい配慮が、創造するよろこびを
さらに高めてくれるでしょう。

Ammusement

遊びへのこだわり

X68の能力の高さを端的に示す
アミューズメントフィールド。
マインドをきわめたゲームフリークの
熱い期待に応える。
画像の美しさが感性を刺激する、
たとえばひと味違う大魔界村なら、
キミのこだわり度は今、全開!

© CAPCOM1991、1993 ALL RIGHTS RESERVED

X68030 / X68000
32bit PERSONAL WORKSTATION / PERSONAL WORKSTATION ·XVI

X68030 [本体+キーボード+マウス・トラックボール]
130㎜FD(5.25型)タイプ CZ-500C-B(チタンブラック) 標準価格398,000円(税別)・〈HD内蔵〉CZ-510C-B(チタンブラック) 標準価格488,000円(税別)

X68030 Compact [本体+キーボード+マウス]
90㎜FD(3.5型)タイプ CZ-300C-B(チタンブラック) 標準価格388,000円(税別)

X68000 XVI Compact [本体+キーボード+マウス]
90㎜FD(3.5型)タイプ CZ-674C-H(グレー) 標準価格298,000円(税別)

●ディスプレイは別売です。●消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は、標準価格には含まれておりません。●画面はハメコミ合成です。

■お問い合わせは… シャープ株式会社 電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

T1002179100900 雑誌 02179-10